

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Ingegneria e Architettura
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

**Progettazione di
una piattaforma Cloud
per analisi predittive
in ambito industria 4.0**

Tesi di laurea in
BIG DATA

Relatore
Dott. Enrico Gallinucci

Candidato
Vlad Mattiussi

Quarta Sessione di Laurea
Anno Accademico 2021-2022

Sommario

Negli ultimi anni c'è stato un grande sviluppo delle tecnologie machine learning e cloud, le quali hanno permesso di semplificare ed efficientare le attività di analisi ed elaborazione di una grande quantità di dati. Molte aziende hanno già cominciato ad adottare queste tecnologie. Il progetto di questa tesi nasce con lo scopo di aiutare un'azienda ad entrare in questo mondo in modo da aumentare la propria competitività rispetto alla concorrenza. Il progetto si pone come obiettivo quello di progettare ed implementare una piattaforma cloud-based in grado di fornire informazioni utili all'azienda ottenute dall'elaborazione dei dati generati dall'apparato di produzione industriale. Questo aiuterà l'azienda a prendere decisioni in maniera tempestiva e supportata dai dati. Il progetto è stato svolto utilizzando tecnologie nuove come il cloud, il low code e alcune tecniche di machine learning. Si è progettata e implementata un'architettura in cloud basata sui servizi offerti da AWS in grado di raccogliere, estrarre e salvare i dati in un sistema di archiviazione, chiamato S3. Da S3, attraverso alcune API i dati passano alla parte di elaborazione e utilizzo di algoritmi per generare le predizioni. Infine i risultati vengono restituiti e mandati di nuovo su S3. Poi, in un secondo momento è stata realizzata, tramite la piattaforma low-code Wavemaker, la Dashboard, cioè la parte di interfaccia visiva che si occupa di far partire le varie interrogazioni e analisi sul database e restituire i risultati in una tabella o grafico e rappresentarli graficamente.

Un grazie a chi mi è stato vicino in questo percorso

Indice

Sommario	iii
1 Definizione del problema	3
1.1 Industria 4.0	3
1.2 Metodologia di sviluppo	7
2 Stato dell'Arte	9
2.1 Big data e industria 4.0	9
2.1.1 Casi d'uso dei Big Data nell'industria 4.0	9
2.1.2 Architettura Big data	10
2.1.3 Piattaforme Big data	11
2.1.4 On-premises vs on cloud	11
2.1.5 Tipi di piattaforme Big data on cloud	13
2.2 Low code e No code	15
2.2.1 Struttura di una piattaforma Low Code	17
2.2.2 Limitazioni	17
2.2.3 Differenze low code e no code	18
2.2.4 Low Code e Intelligenza Artificiale	19
3 Architettura implementata	21
3.1 Architettura funzionale	22
3.2 Architettura tecnologica	25
3.2.1 back-end	26
3.2.2 Front-end	29
3.3 Implementazione	30
3.3.1 Modelli predittivi	31
3.3.2 Front-end	34
4 Test	35
5 Conclusioni	41

Elenco delle figure

1.1	Industria 4.0	5
1.2	Differenze programmazione low code e tradizionale	7
2.1	Architettura big data	10
2.2	Livelli cloud	13
2.3	Il cloud computing	15
3.1	Architettura funzionale	23
3.2	Esempio architettura mqtt	24
3.3	Architettura tecnologica back-end	26
3.4	AWS IoT Core	27
3.5	Schermata raffigurante l'interfaccia di wavemaker	30
3.6	Esempio di una semplice rete neurale MLP	32
4.1	Grafico della macchina 1 con corrispondenza fra train e test set	36
4.2	Grafico della macchina 1 mostrante la predizione su N periodi successivi	36
4.3	Grafico della macchina 2 con corrispondenza fra train e test set	37
4.4	Grafico della macchina 2 mostrante la predizione su N periodi successivi	38
4.5	Grafico della macchina 1 mostrante la predizione effettuata con Sarimax	38
4.6	Grafico della macchina 2 mostrante la predizione effettuata con Sarimax	39
4.7	Risultati mostrati sull'applicazione creata con Wavemaker	40

Elenco dei listati

Introduzione

Negli ultimi anni c'è stato un grande sviluppo delle tecnologie machine learning e cloud, facilitando e velocizzando l'analisi, la conservazione e l'utilizzo di una grande quantità di dati. Molte aziende hanno cominciato ad adottare queste tecnologie, o in modalità autonoma, o affidandosi ad altre aziende più competenti, al fine di sfruttarle al meglio queste nuove tecnologie. Tramite l'utilizzo di queste, le aziende riescono ad avere informazioni utili al fine di aiutarle a compiere scelte intelligenti; scelte e decisioni che serviranno a contribuire ad un miglioramento e accrescimento di essa. Il progetto di questa tesi nasce proprio da questo: aiutare un'azienda ad "entrare" in questo mondo e rendersi più competitiva rispetto alla concorrenza. Infatti, il progetto si pone come obiettivo quello di progettare ed implementare una piattaforma cloud-based in grado di fornire informazioni utili all'azienda, come ad esempio predizioni riguardo ai pezzi prodotti da alcune macchine nella fabbrica sulla catena di produzione. Questo aiuterà l'azienda a portarsi avanti con i ritmi e prendere le giuste decisioni gestionali. Il progetto è stato svolto utilizzando tecnologie nuove come il cloud e le reti neurali in campo machine learning. Le informazioni predittive sono gestite tutte in cloud, tramite l'utilizzo di alcuni servizi offerti dalla piattaforma AWS. Mentre l'applicazione è stata creata utilizzando "Wavemaker", una piattaforma in cloud che permette di creare applicazioni utilizzando poco o zero codice; questo velocizza e semplifica molto il processo produttivo. La tesi è composta dai seguenti capitoli:

- Capitolo 1: inizialmente si affronta una breve introduzione sul mondo dell'industria 4.0. per capire e studiare le ultime necessità e obiettivi nel moderno mondo industriale. Viene inoltre introdotta la metodologia di sviluppo tramite l'utilizzo di piattaforme low-code e no-code, ponendole a confronto con le tradizionali metodologie.
- Capitolo 2: viene esposto lo stato dell'arte per quanto riguarda i progetti e architettura big data, passando per le diverse tematiche che toccano da vicino il progetto di questa tesi. Quindi si parla delle differenze tra avere un'infrastruttura on-premise o su cloud, dei vari livelli di utilizzo delle varie piattaforme in cloud, sia dal punto di vista dell'hardware che del software.

Infine, si espone la tecnologia recente del low-code e low-code, analizzandole anche in relazione ai temi big data e machine learning.

- Capitolo 3: qui viene mostrata l'architettura del progetto sviluppato. Esso si divide in due parti principali: back-end e front-end. Il back-end end rappresenta la parte principale, comprende l'analisi dei dati, gli algoritmi predittivi e di come il cloud è stato utilizzato nei vari passaggi di dati. La parte front-end invece, comprende lo sviluppo dell'applicazione, quindi anche della parte di interfaccia grafica, realizzata con Wavemaker. In un primo momento viene illustrata l'architettura funzionale, cioè vengono spiegate, componente per componente, le varie funzioni che si devono svolgere nei vari passaggi. Successivamente viene spiegata l'architettura tecnologica, cioè si vanno ad esplorare i vari servizi sfruttati per adempiere le funzioni dell'architettura funzionale. Infine, viene spiegato in linea di massima come sono stati implementati i vari componenti dell'architettura andando più nel dettaglio.
- Capitolo 4: sono riportati i risultati dei test svolti sull'architettura con diverse configurazioni e in diverse condizioni.
- Capitolo 5: inserito a completamento della tesi, è dedicato alle conclusioni e ai possibili sviluppi futuri.

Capitolo 1

Definizione del problema

Il progetto è nato dall'esigenza di fornire delle tecnologie e strumenti ad un'azienda per facilitarne l'entrata nell'industria 4.0. Questa azienda possiede una fabbrica di bombole ad ossigeno. Si vuole procedere attraverso l'installazione di alcuni dispositivi su alcune macchine presenti sulla catena di montaggio della fabbrica: questi dispositivi raccoglieranno informazioni riguardo alla quantità di elementi prodotti entro un dato intervallo di tempo. Poi queste informazioni saranno elaborate da alcuni algoritmi. Alla fine, queste informazioni saranno visualizzate tramite un'applicazione e permetterà all'azienda di vedere predizioni sull'andamento futuro della produzione sulla catena di montaggio. Queste preziosi informazioni saranno utilizzate dai decisori aziendali per definire e supportare le strategie di business, al fine di compiere le migliori scelte con l'obiettivo di trarre vantaggi competitivi, migliorare le prestazioni operative e la profittabilità e, più in generale, creare valore per l'azienda. Il progetto in questione permette all'azienda di rispondere in modo adeguato a molti di questi quesiti dei processi aziendali. Un altro punto che si vuole raggiungere è l'utilizzo e sperimentazione della tecnologia low code, una nuova metodologia di sviluppo software che verrà spiegata successivamente.

1.1 Industria 4.0

L'industria 4.0 si riferisce a una nuova fase della rivoluzione industriale che si concentra fortemente sull'interconnettività, l'automazione, l'apprendimento automatico e i dati in tempo reale. L'Industria 4.0, che comprende argomenti come l'IoT, intelligenza artificiale, il cloud computing, l'unione tra produzione fisica e digitalizzazione, l'apprendimento automatico e i big data per creare un ecosistema meglio connesso per le aziende che si concentrano sulla produzione e sulla gestione della supply chain. Quindi, l'Industria 4.0 è una visione ampia con framework e architetture di riferimento, caratterizzata principalmente dal ponte tra elementi

industriali fisici e tecnologie digitali nei cosiddetti sistemi cyber-fisici. Sebbene ogni azienda e organizzazione che opera oggi sia diversa, tutte affrontano una sfida comune: la necessità di connessione e accesso a informazioni dettagliate in tempo reale su processi, partner, prodotti e persone. Inoltre, l'Industria 4.0 non riguarda solo l'investimento in nuove tecnologie e strumenti per migliorare l'efficienza produttiva, ma anche rivoluzionare il modo in cui l'intera azienda opera e cresce. Infatti, l'Industria 4.0 è una trasformazione ad alta intensità di informazioni della produzione (e delle industrie correlate) in un ambiente connesso, per passare ad un successivo sfruttamento di dati utilizzabili come modo e mezzo per realizzare un'industria intelligente. Queste fabbriche intelligenti sono dotate di sensori avanzati, software integrati e robotica che raccolgono e analizzano i dati e consentono un migliore processo decisionale. Un valore ancora maggiore viene creato quando i dati delle operazioni di produzione vengono combinati con i dati operativi provenienti da ERP, supply chain, servizio clienti e altri sistemi aziendali per creare livelli completamente nuovi di visibilità e approfondimenti da informazioni precedentemente isolate. Queste tecnologie digitali portano a una maggiore automazione, manutenzione predittiva, auto-ottimizzazione dei miglioramenti dei processi e, soprattutto, un nuovo livello di efficienza e reattività ai clienti non possibile in precedenza. Con un investimento minimo, il personale addetto al controllo qualità può configurare uno smartphone connesso al cloud per monitorare i processi di produzione praticamente da qualsiasi luogo. Applicando algoritmi di apprendimento automatico, i produttori possono rilevare gli errori immediatamente, piuttosto che nelle fasi successive quando i lavori di riparazione sono più costosi.

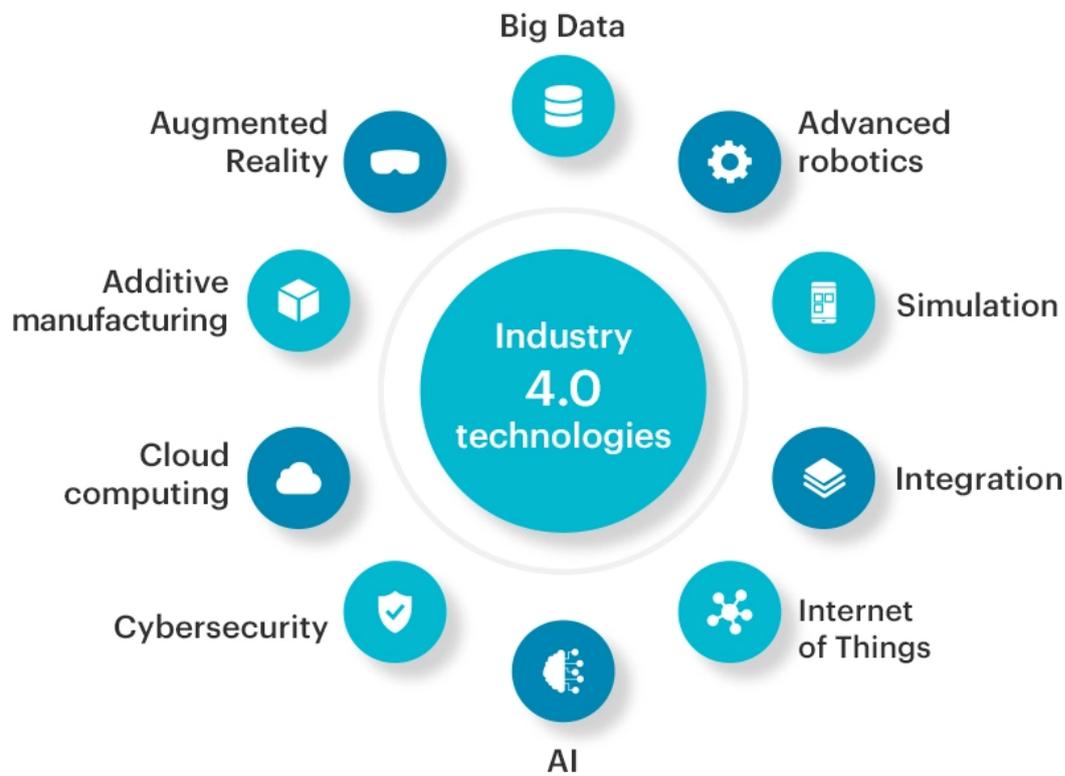


Figura 1.1: Industria 4.0

Di seguito le caratteristiche più importanti dell'industria 4.0:

- L'Internet of Things (IoT) è una componente chiave di queste industrie "intelligenti". Le macchine in fabbrica sono fornite di sensori dotati di un indirizzo IP che consente alle macchine di connettersi con altri dispositivi che hanno accesso al Web. Questa meccanizzazione e connettività consente di raccogliere, analizzare e scambiare grandi quantità di dati utili.
- Il cloud computing è una pietra miliare di qualsiasi strategia Industry 4.0. La piena realizzazione della produzione intelligente richiede connettività e un'integrazione fra ingegneria, supply chain, produzione, vendita, distribuzione e assistenza. Il cloud aiuta a renderlo possibile. Inoltre, la grande quantità di dati archiviati e analizzati può essere elaborata in modo più efficiente ed economico. Il cloud computing può anche ridurre i costi di avvio per i produttori di piccole e medie dimensioni che possono ridimensionare le loro esigenze e scalare man mano che la loro attività cresce.

- L'intelligenza artificiale e il machine learning consentono alle aziende manifatturiere di sfruttare appieno il volume di informazioni generate non solo in fabbrica, ma anche attraverso le loro ulteriori unità aziendali e persino da partner e fonti di terze parti. L'intelligenza artificiale e il machine learning possono creare informazioni dettagliate fornendo visibilità, prevedibilità e automazione delle operazioni e dei processi aziendali. Ad esempio: le macchine industriali sono soggette a guasti durante il processo di produzione. L'utilizzo dei dati raccolti da queste risorse può aiutare le aziende a eseguire la manutenzione predittiva basata su algoritmi di machine learning, con conseguente maggiore uptime ed efficienza.
- Le esigenze delle operazioni di produzione in tempo reale significano che alcune analisi dei dati devono essere eseguite sul "limite" ("edge"), ovvero dove vengono creati i dati. Ciò riduce al minimo il tempo di latenza da quando i dati vengono prodotti a quando è richiesta una risposta. Ad esempio, il rilevamento di un problema di sicurezza o di qualità può richiedere un'azione quasi in tempo reale con l'apparecchiatura. Il tempo necessario per inviare i dati al cloud aziendale e quindi tornare in fabbrica potrebbe essere troppo lungo e dipende dall'affidabilità della rete. L'utilizzo dell'edge computing significa anche che i dati rimangono vicino alla loro fonte, riducendo i rischi per la sicurezza.
- Le aziende manifatturiere non hanno sempre considerato l'importanza della sicurezza informatica o dei sistemi cyber-fisici. Tuttavia, la stessa connettività delle apparecchiature operative in fabbrica o sul campo (OT) che consente processi di produzione più efficienti espone anche nuove vulnerabilità di sfruttamento per attacchi dannosi e malware. Quando si affronta una trasformazione digitale verso l'Industria 4.0, è essenziale considerare un approccio alla sicurezza informatica che comprenda apparecchiature IT e OT.
- La trasformazione digitale offerta dall'Industria 4.0 ha permesso ai produttori di creare gemelli digitali che sono repliche virtuali di processi, linee di produzione, fabbriche e supply chain. Un gemello digitale viene creato estraendo dati da sensori, dispositivi, PLC e altri oggetti IoT connessi a Internet. I produttori possono utilizzare i gemelli digitali per aumentare la produttività, migliorare i flussi di lavoro e progettare nuovi prodotti. Simulando un processo di produzione, ad esempio, i produttori possono testare le modifiche al processo per trovare modi per ridurre al minimo i tempi di fermo o migliorare la capacità.

1.2 Metodologia di sviluppo

L'obiettivo è duplice: da un lato si vuole sviluppare un'applicazione che consenta all'azienda di entrare nell'industria 4.0; dall'altro si vuole sviluppare il progetto in modo migliore possibile. Per questo si è deciso di utilizzare tecnologie recenti come il cloud computing e machine learning (spiegati precedentemente), ma anche di fare affidamento a tecnologie low code/no code, che oggi si pongono in alternativa alla programmazione tradizionale. Come dice la parola, la programmazione tradizionale è quella utilizzata per la maggior parte nel mondo ed è anche quella che impariamo all'università. È un processo manuale per il quale uno sviluppatore crea da zero una qualsiasi applicazione con un particolare linguaggio di programmazione. La programmazione tradizionale si riferisce alla collaborazione di un intero team di sviluppatori e programmatori per raccogliere requisiti specifici, sviluppare un piano e collaborare con un team di sviluppo per creare del codice personalizzato per un'applicazione per soddisfare le esigenze specificate. Sebbene l'approccio tradizionale sia completamente valido e un metodo assolutamente accettabile, questi progetti sono spesso complessi, costosi e vengono ritardati a causa di molteplici fattori quali: potenziali errori di codice nel software, stime inaccurate, difficoltà in fase di testing e vari potenziali ritardi.

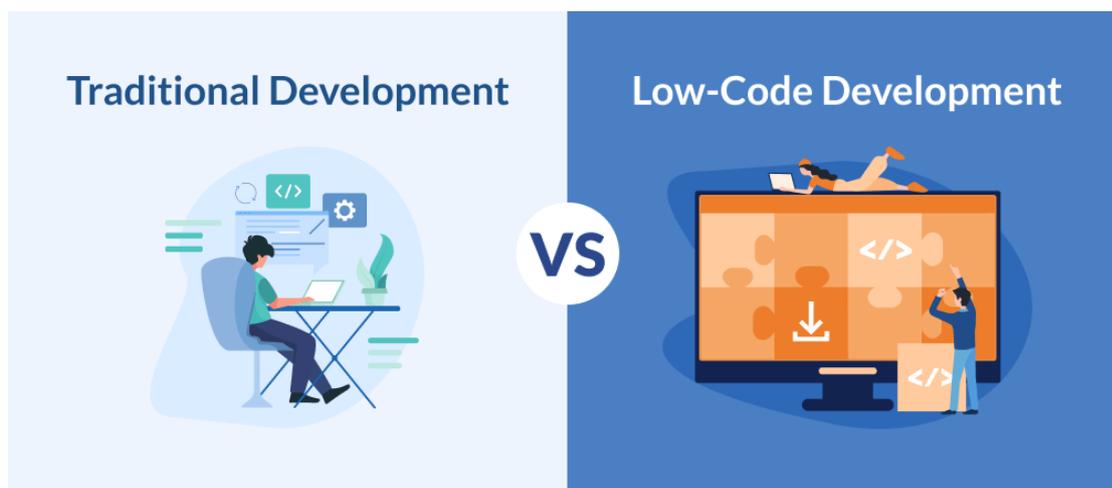


Figura 1.2: Differenze programmazione low code e tradizionale

Inoltre, un approccio di sviluppo tradizionale richiede un ciclo di manutenzione continuo da parte dello sviluppatore per mantenere l'applicazione sicura e aggiornata. Lo sviluppo tradizionale richiede competenze tecniche, a differenza del low code. È necessario assumere sviluppatori web professionisti per programmare e

progettare manualmente la soluzione software: la qualità del prodotto sarà alta e tutto è costruito secondo le proprie richieste, nonostante ciò richiedono un investimento considerevole di tempo, denaro e risorse. Lo sviluppatore segue tutte le fasi del ciclo di vita di un'applicazione, dalla raccolta dei requisiti e dalla ricerca al test e alla distribuzione. Mentre nello sviluppo low code, questo processo è automatizzato ed è più rapido. Quindi, mentre lo sviluppo di software tradizionale si basa sulla scrittura di codice con l'uso di linguaggi di programmazione, il low-code adotta un approccio di sviluppo visivo. Utilizza componenti drag-and-drop, logica basata su modelli e un'interfaccia grafica user-friendly per permettere e facilitare lo sviluppo. Questo approccio consente agli sviluppatori di tutti i livelli di creare applicazioni. Inoltre, il low-code automatizza gran parte del ciclo di distribuzione del software. Tramite tutte queste agevolazioni, gli sviluppatori possono concentrare la maggior parte dei loro sforzi sulla fornitura di un reale valore aziendale.

Capitolo 2

Stato dell'Arte

Come già accennato precedentemente, si vuole conoscere e utilizzare le ultime tecnologie sviluppate in ambito aziendale e industriale. Tra queste sicuramente il tema dei Big Data è un elemento importante da trattare.

2.1 Big data e industria 4.0

L'industria 4.0 abbraccia e sfrutta pienamente la rivoluzione digitale data dall'avvento di tecnologie big data e intelligenza artificiale. I produttori nel mercato di oggi cercano di raggiungere l'eccellenza produttiva attraverso la modellazione, l'analisi e la condivisione di grandi quantità dati.

2.1.1 Casi d'uso dei Big Data nell'industria 4.0

La quantità di informazioni prodotte dai dispositivi legati all'IoT e dai sistemi di produzione odierni deve essere tradotta in idee attuabili. Ecco perché i Big Data classificano le informazioni raccolte e traggono conclusioni intelligenti che aiutano a migliorare le operazioni delle aziende. Di seguito alcuni esempi concreti.

- Miglioramento della gestione dei magazzini: grazie a sensori e dispositivi portatili, le aziende possono migliorare l'efficienza operativa rilevando errori umani, eseguendo controlli di qualità e mostrando percorsi di produzione o assemblaggio ottimali.
- Pianificazione intelligente: previsioni più accurate e significative grazie alla visualizzazione dell'attività attraverso analisi di feedback utenti e tendenze di mercato oltre ai dati storici. Ciò consente all'azienda di modificare/ottimizzare il proprio portafoglio prodotti.

- **Manutenzione predittiva:** i sensori alimentati dai dati identificano possibili guasti nel funzionamento dei macchinari prima che si trasformino in un guasto, identificando i problemi nei modelli. Il sistema invia un avviso all'apparecchiatura in modo che possa reagire in tempo.

Data la particolarità del problema e la natura data-intensive delle operazioni sorge la necessità di creare un'architettura appositamente progettata.

2.1.2 Architettura Big data

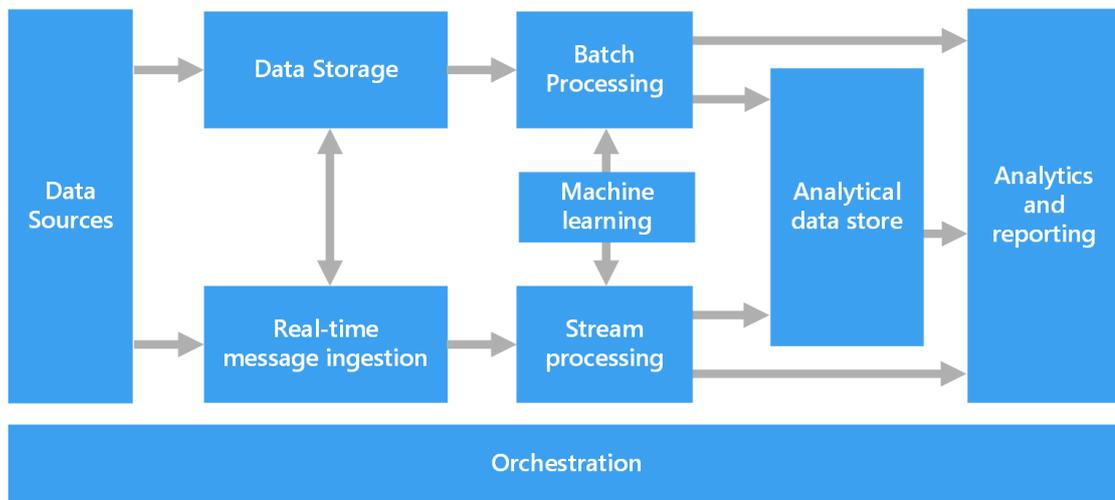


Figura 2.1: Architettura big data

Il diagramma sopra mostra una tipica composizione di un'architettura big data, non è detto che tutte le soluzioni contengano questi componenti; di seguito si descrivono i componenti più significativi, utilizzati anche nel progetto.

- **Sorgente dati:** la componente da cui tutti i dati provengono, in questo caso dai dispositivi installati in fabbrica sulle macchine produttrici.
- **Storage dati:** i dati per le operazioni di elaborazione vengono in genere archiviati in un archivio di file distribuito che può contenere volumi elevati di file di grandi dimensioni in vari formati. Nel caso del progetto si è utilizzato AWS S3 come principale metodo di archiviazione.
- **Machine learning:** la componente contenente gli algoritmi di elaborazione e predizione dei dati. Nel progetto questo ruolo può essere svolto da uno script python oppure dal framework AWS Forecast.

2.1.3 Piattaforme Big data

Ormai le aziende raccolgono grandi quantità di dati per consentire analisi avanzate e il flusso costante di informazioni provenienti da varie fonti sta diventando sempre più intenso, soprattutto con il progresso della tecnologia. Ed è qui che entrano in gioco le piattaforme di big data per archiviare e analizzare questa massa sempre crescente di informazioni. Una piattaforma Big Data è una soluzione di elaborazione integrata che combina numerosi sistemi software, strumenti e hardware per la gestione dei big data. Si tratta di un'architettura che risolve tutte le esigenze di dati di un'azienda indipendentemente dal volume e dalle dimensioni dei dati. Grazie alla loro efficienza nella gestione dei dati, le aziende stanno adottando sempre più piattaforme di big data per raccogliere dati e convertirli in informazioni aziendali strutturate e fruibili. Queste piattaforme possono essere fornite in due modi: on premise e on cloud.

2.1.4 On-premises vs on cloud

Con il termine software on-premises si fa riferimento alla fornitura di servizi informatici installati e gestiti attraverso computer locali. Questo significa che l'utente deve gestire autonomamente l'installazione e la manutenzione sia della parte hardware delle macchine, sia della parte riguardante il software. Il concetto on-premises si contrappone all'erogazione di servizi software off-premises, in modalità SAAS o in Cloud computing, dove la fruizione del programma avviene attraverso l'accesso a un computer (o a un'architettura di hardware) in remoto.

Entrambi questi approcci si adattano a diversi casi d'uso. Quando si considerano i pro e i contro on-premise rispetto al cloud, ci sono alcune aree principali da considerare:

- **Scalabilità:** i data warehouse crescono abbastanza rapidamente nel tempo, il che significa che ci sarà necessità di espandere regolarmente lo storage disponibile. Con i data warehouse on-premise, si dovrà aggiungere spazio di archiviazione al data warehouse. Ciò significa acquistare e configurare hardware di archiviazione aggiuntivo. Se invece si ha bisogno di ridimensionare, si potrebbe finire con dischi rigidi indesiderati. Nei data warehouse cloud è possibile aumentare la scalabilità modificando il livello di sottoscrizione. Il fornitore di servizi può allocare tutto lo spazio di cui si ha bisogno. In genere non è necessario apportare modifiche alla configurazione, anche se il costo annuale aumenterà. I provider di servizi cloud spesso consentono di ridimensionarsi con la stessa facilità.

- **Costi:** il data warehousing basato sul cloud, al contrario della modalità on premise, elimina la maggior parte dei costi iniziali. Inoltre, si paga solo per le risorse utilizzate, il che migliora l'efficienza.
- **Velocità:** le soluzioni cloud possono aggiungere un certo grado di latenza alle transazioni e, poiché il Data Warehouse si trova al di fuori della rete locale, le richieste avvengono alla stessa velocità di altre transazioni Internet. Se l'intera azienda si trova in un'unica posizione, un Data Warehouse on-premise sarà la scelta migliore. Tuttavia, se il fornitore di cloud rende disponibile il cloud su più server sparsi per tutto il mondo è possibile che i servizi cloud migliorino la velocità complessiva.
- **Affidabilità:** spesso l'affidabilità e la disponibilità dei servizi sono i principali problemi delle infrastrutture. I Data Warehouse basati su cloud offrono un accordo sul livello di servizio che garantisce una certa disponibilità. Ad esempio, Amazon promette un tempo di attività minimo del 99,99% di disponibilità per il suo servizio EC2. Google promette una percentuale di uptime mensile del 99,9% per Cloud Storage e BigQuery. Google, Amazon e altri provider in cloud possono replicare i dati degli utenti su più cluster per garantire la massima affidabilità. Per i data warehouse on-premise, il tempo di attività è una propria responsabilità. Si dovrà investire in un hardware affidabile e un team di supporto in grado di risolvere i problemi giorno e notte.
- **Sicurezza:** nella maggior parte dei casi, le soluzioni cloud sono più sicure delle soluzioni on-premise nella maggior parte dei casi d'uso. Questo può sembrare contro intuitivo, poiché le soluzioni cloud inviano dati sensibili a terzi, mentre le soluzioni on-premise mantengono tutto all'interno della rete aziendale. Tuttavia i dati raramente rimangono all'interno della rete aziendale; potrebbe essere necessario trasmettere informazioni a partner come commercialisti o consulenti legali, oppure potrebbe essere necessario fornire una copia dei dati ai revisori. Ma il rischio più comune per i dati è il proprio personale. I dipendenti spesso violano la politica dei dati copiando i dati sensibili sul proprio portatile o su una chiavetta USB in modo che possano continuare a lavorare da casa. Se perdono uno di questi dispositivi, si potrebbe dover incorrere in una potenziale violazione dei dati. Le soluzioni cloud adottano un approccio basato sulla sicurezza per le transazioni di dati esterni. Ad esempio, Google BigQuery e Amazon Redshift dispongono entrambi di funzionalità di sicurezza per garantire la sicurezza dei dati in ogni punto del loro percorso. I dipendenti possono accedere in remoto ai dati tramite un canale approvato. I Data Warehouse on-premise sono l'opzione più sicura se supportata da una rigida politica di accesso ai dati. Ma l'archiviazione cloud

offre una sicurezza flessibile che mantiene i dati al sicuro nella maggior parte delle situazioni.

2.1.5 Tipi di piattaforme Big data on cloud

Le piattaforme possono essere fornite come servizio attraverso diversi "livelli" di cloud, cioè è possibile gestire solo una parte di elementi della piattaforma, mentre il resto sarà responsabilità del fornitore. Anche le piattaforme low code fanno parte di questa categoria, in particolare nel "range" da PaaS a SaaS, variando da diversa azienda ad azienda produttrice di piattaforme low code. Infatti, ad esempio, la piattaforma SaaS consente agli utenti di costruire un'applicazione senza scrivere codice (o comunque poco) con l'uso di un'interfaccia grafica che consente agli utenti di trascinare e rilasciare componenti modulari e riutilizzabili per creare la propria applicazione.

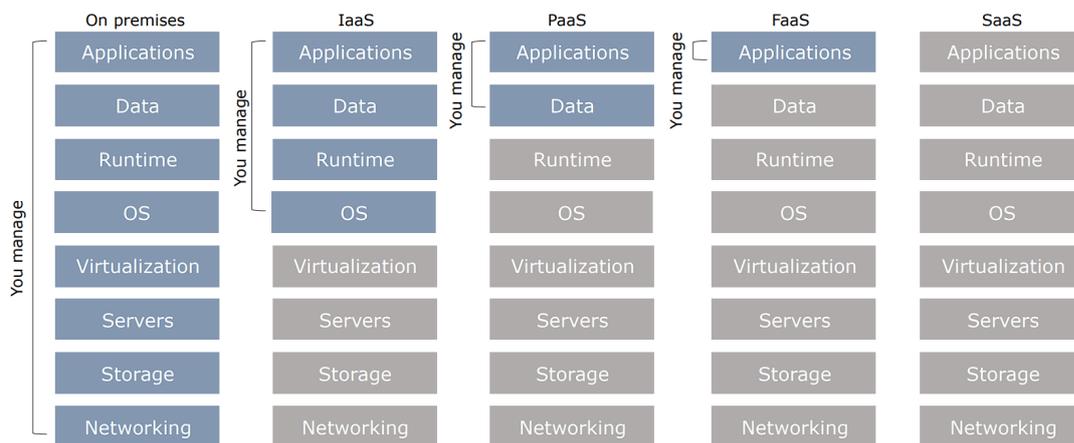


Figura 2.2: Livelli cloud

- IaaS, Infrastructure as a Service: è un modello di fornitura di servizi in cui un'organizzazione colloca "al di fuori" le apparecchiature utilizzate, questo include l'archiviazione delle informazioni, l'hardware, i server e i componenti di rete. IaaS è talvolta indicato anche come "Hardware as a Service" o HaaS. Il vantaggio più evidente dell'utilizzo di un IaaS, è quello di trasferire al fornitore problemi di gestione relativi alle apparecchiature informatiche. Un altro vantaggio è la riduzione dei costi, come generalmente nelle tecnologie associate al cloud computing, per pagare solo per ciò che viene usato. IaaS

consente inoltre una scalabilità praticamente automatica e trasparente per il consumatore, lasciando la responsabilità ai fornitori di servizi.

- PaaS, Platform as a Service: con il cloud computing e la sua rapida crescita è stato introdotto il PaaS. Le applicazioni PaaS sono note anche come soluzioni on-demand basate sul Web o SaaS. Il provider, oltre a risolvere i problemi nell'infrastruttura hardware, gestisce anche il software. Non è necessario installare, configurare o mantenere un sistema operativo, database o server, dato che questi sono forniti dalla piattaforma. Una piattaforma distribuita come PaaS risolve più problemi rispetto a una soluzione che fornisce solo un'infrastruttura distribuita come IaaS, poiché presenta molte limitazioni correlate all'ambiente di runtime. Tra questi ci sono il tipo di sistema, il linguaggio di programmazione e il gestore di database. Aziende come eBay, Google, iTunes e YouTube sono tra quelle che utilizzano questo modello.
- Faas, Function as a Service: permette di eseguire un determinato codice e quindi una funzionalità in risposta ad un determinato evento senza alcuna infrastruttura da gestire o da configurare per rendere operativa tale funzionalità. Quindi, è un modello di esecuzione basato su eventi, che viene eseguito in container stateless; le funzioni gestiscono la logica e lo stato sul lato server utilizzando i servizi di un provider FaaS. Un esempio di tecnologia Faas è AWS Lambda (utilizzato anche nel progetto).
- SaaS, Software as a Service: è il più noto dei livelli di cloud computing. SaaS è un modello di distribuzione software che fornisce ai clienti l'accesso ad esso attraverso la rete (internet). Pertanto, non devono preoccuparsi della configurazione, della distribuzione e della manutenzione delle applicazioni, perché tutte queste attività diventano responsabilità del fornitore. Le applicazioni distribuite attraverso questo modello possono raggiungere qualsiasi azienda indipendentemente dalle dimensioni o dalla posizione geografica. Un esempio di SaaS può essere descritto come la classica applicazione utilizzata direttamente sul browser web (ad esempio l'email).

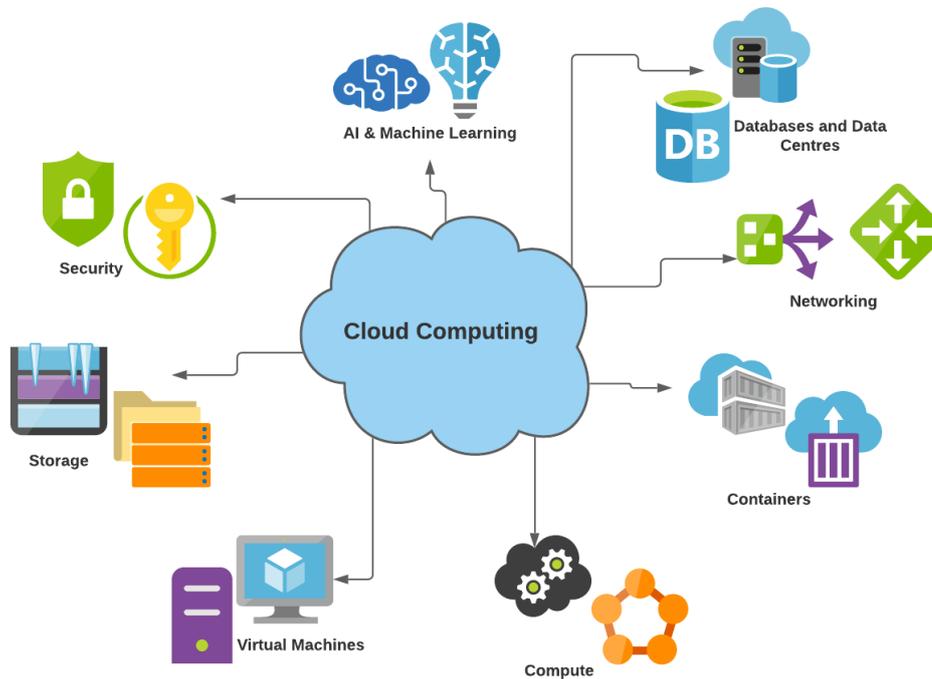


Figura 2.3: Il cloud computing

Il cloud computing è rappresentato da una crescita graduale. Nonostante sia una tecnologia già utilizzata da alcuni anni, rimane ancora da "assorbire" pienamente come tendenza centrale nelle organizzazioni. Il livello di accoglienza del cloud tra le aziende varia con le dimensioni di questi: le medie e piccole imprese iniziano già a implementare queste soluzioni, mentre le grandi aziende lo fanno in base a particolari esigenze. Dall'altra parte, ci sono gli utenti finali, ai quali il cloud computing ha cambiato il modo in cui svolgono le loro attività, migliorando nella maggior parte dei casi e consentendo loro di collaborare con altri utenti in luoghi diversi.

2.2 Low code e No code

Molte piattaforme low code, tra cui quella utilizzata nel progetto studiato (Wavemaker), sono presenti in cloud. In questa sezione si parlerà del low code in generale.

Low code (o no code) sono degli approcci di sviluppo software che richiedono poco o zero scrittura di codice per costruire applicazioni e processi. È un approccio che permette di velocizzare l'intero processo di sviluppo e accelerare le consegne. Permette di creare software applicativi attraverso moduli di configurazioni e interfacce grafiche, invece di programmare scrivendo codice. È sufficiente descrivere come l'applicazione stessa deve essere fatta e quali funzionalità dovrà svolgere. Ciò è possibile utilizzando, direttamente attraverso l'interfaccia della piattaforma low code, elementi grafici, tecniche visuali e dichiarative, ed eseguendo operazioni "drag and drop": in questo modo l'applicazione si può sviluppare più velocemente, perché è l'intelligenza dell'algoritmo che poi si preoccupa di tradurla in codice. La piattaforma Low Code può focalizzarsi sulla progettazione e sullo sviluppo di un particolare tipo di applicazione: database, processi aziendali o interfacce utente, come le applicazioni web, applicazioni mobile. Tali piattaforme possono produrre applicazioni interamente operative o richiedere modifiche per sviluppare funzionalità aggiuntive dell'applicativo generato. Uno dei vantaggi è quello di permettere a persone senza particolari competenze tecniche di programmazione di contribuire alle fasi iniziali di sviluppo di un'applicazione. Dunque, le piattaforme low code non soltanto aiutano gli sviluppatori professionisti ad accelerare la velocità di programmazione, ma estendono l'adozione dei propri strumenti anche agli esperti del business che, senza necessità di scrivere codice, diventano in grado di dare il loro contributo e partecipare allo sviluppo dell'applicazione enterprise. Le piattaforme di sviluppo low-code riducono il costo iniziale di installazione, implementazione e formazione. Oltre a supportare la scalabilità per le applicazioni enterprise, lo sviluppo low code può soddisfare tre esigenze chiave:

- **Velocità:** accelerando il ritmo di sviluppo, il metodo low code soddisfa l'esigenza di rapidità in azienda. Numerose aziende hanno adottato una piattaforma o uno strumento low code per poter migliorare le funzionalità IT esistenti, innovare prodotti e servizi e diventare più agili, in modo da accelerare il "time-to-market".
- **Affidabilità:** le aziende con la più ridotta tolleranza in termini di downtime e perdita di dati, e con i più stringenti requisiti in fatto di auditing, sono quelle con maggior probabilità di eseguire applicazioni su una piattaforma low code.
- **Complessità logica di business:** oggi molte organizzazioni utilizzano codice custom per sviluppare applicazioni destinate a gestire logica di business complessa: ma in futuro esse probabilmente implementeranno ed eseguiranno tali applicazioni usando tecniche low code, invece di sviluppare codice custom.

2.2.1 Struttura di una piattaforma Low Code

Una piattaforma di sviluppo low code (LCDP) ha in genere 3 componenti:

- Interfaccia utente grafica (GUI) per la programmazione: questa interfaccia drag and drop consente agli utenti aziendali di definire input e output e le operazioni necessarie
- Integrazioni per I/O: la maggior parte dei dati aziendali viene archiviata in database. Gli LCDP offrono interfacce per i database. La maggior parte delle applicazioni aziendali espone il proprio output tramite API, siti Web o app. Gli LCDP consentono agli utenti di creare queste funzionalità di output con facilità
- Un gestore di applicazioni: gli LCDP forniscono strumenti per creare, eseguire il debug, distribuire e gestire applicazioni low-code in modo che gli utenti possano tenerle aggiornate. Utilizzando questi componenti, gli utenti aziendali possono creare prodotti MVP (Minimum Viable Products) per soddisfare le loro esigenze urgenti. In futuro, se questi programmi dovranno essere drasticamente aggiornati, gli utenti più esperti potranno esaminare gli MVP per comprendere le funzionalità del programma e crearne una versione aggiornata, se necessario.

2.2.2 Limitazioni

Come accennato nel capitolo precedente, riguardo la programmazione tradizionale, sebbene lo scopo dello sviluppo low-code sia quello di facilitare il processo di sviluppo del software, esso non è adatto a tutte le attività di sviluppo a causa di alcune limitazioni.

- Opzioni di personalizzazione e flessibilità limitate: gli strumenti di personalizzazione sono limitati negli LCDP che costringono le aziende a modificare i processi aziendali per soddisfare le funzionalità delle applicazioni low-code. Le app LCDP hanno componenti limitati. Quando un'organizzazione desidera aggiungere una funzionalità specifica, i blocchi predefiniti "drag and drop" della selezione potrebbero non soddisfare il requisito. In questi casi, l'azienda deve scrivere codice custom per abilitare particolari funzionalità. E questi codici aggiuntivi all'interno degli LCD possono essere trascurati nei momenti di manutenzione, portando a interruzioni o bug imprevisti.
- Opzioni di integrazione limitate: quando le imprese sviluppano un'app utilizzando un LCDP, possono riscontrare problemi di integrazione, soprattutto quando desiderano eseguire la migrazione di applicazioni legacy.

- **Sicurezza:** gli LCDP sono software basati su cloud, sviluppati da utenti con esperienza limitata nella sicurezza delle informazioni che non hanno il pieno controllo del processo di sviluppo. Pertanto, le violazioni della sicurezza sono un rischio negli LCDP e le applicazioni low code, in particolare quelle esterne, devono essere controllate per problemi di sicurezza.
- **Strategia di lock-in:** l'utente dipende interamente da un fornitore di software che non può passare a un altro fornitore senza notevoli costi di commutazione.
- **Limitazioni alla personalizzazione:** alcune soluzioni di machine learning low-code potrebbero non consentire agli utenti di modificare parametri specifici. Inoltre, le aziende potrebbero dover fare affidamento sull'esperienza di data scientist e data engineer per le attività di elaborazione dei dati.
- **Scalabilità:** al momento, è impossibile creare una soluzione scalabile utilizzando una piattaforma di machine learning low code che risolva un problema complesso.

Quindi, come si può notare, per quanto riguarda lo sviluppo di applicazioni machine learning, al momento di scrittura, questi strumenti non hanno la flessibilità o la manutenibilità delle applicazioni tradizionali. Pertanto, le aziende devono avere una chiara comprensione e visione di quali problemi affrontare utilizzando questi strumenti. Se si aspira a sviluppare soluzioni scalabili, allora il tradizionale approccio di machine learning sarebbe un modo migliore per procedere.

2.2.3 Differenze low code e no code

I termini "low-code" e "no code" sono spesso citati insieme perché sono simili. Non ci siamo imbattuti in una definizione ben definita, quindi per ora abbiamo supposto che siano la stessa cosa. Entrambe le piattaforme low-code e no-code utilizzano interfacce grafiche per consentire agli utenti di sviluppare le proprie soluzioni aziendali anche senza una conoscenza approfondita della programmazione. La principale differenza tra piattaforme low-code e no code, come implicato dai loro nomi, è che le piattaforme low-code possono includere la scrittura manuale di codice, mentre le piattaforme no-code no. Ciò significa essenzialmente che la tecnologia no-code è progettata specificamente per gli utenti meno esperti, mentre la tecnologia low-code si rivolge sia a questi, che agli sviluppatori professionisti. Poiché le piattaforme low-code possono comportare la programmazione, sono in grado di creare applicazioni più grandi e complesse rispetto alle piattaforme senza codice. Per una maggiore flessibilità e controllo sul ciclo di sviluppo, le aziende lungimiranti implementano piattaforme che combinano tecnologie low-code e no-code.

2.2.4 Low Code e Intelligenza Artificiale

Automated Machine Learning

L'automated machine learning (AutoML) è il processo che consente di automatizzare le attività manuali e monotone che i data scientist devono eseguire per creare e addestrare modelli di machine learning. La selezione e l'ingegnerizzazione delle funzionalità, degli algoritmi e l'ottimizzazione degli iperparametri sono esempi di tali attività. Potrebbe capitare di confondere l'AutoML con le soluzioni low/no-code AI. Anche se alla fine potrebbero unirsi e diventare un'unica categoria, attualmente hanno caratteristiche diverse. Da un lato le piattaforme di machine learning no-code consentono agli utenti meno esperti di creare modelli, dall'altro la maggior parte delle soluzioni AutoML mira a consentire ai data scientist di diventare più efficienti. Forniscono inoltre una maggiore trasparenza nell'intera pipeline di machine learning e aiuta a perfezionare il modo in cui vengono creati i modelli.

Differenze con low/no-code IA

Come già scritto precedentemente, l'AutoML è un'attività completamente diversa dagli strumenti low code per la data science. Le applicazioni AutoML sono limitate al solo addestramento di modelli, in una modalità completamente automatizzata. Gli strumenti low code per il data science, al contrario, coprono una vastità di diverse operazioni, dalle trasformazioni dei dati agli algoritmi di machine learning, esponendo molti degli iper-parametri coinvolti nella definizione del modello o della trasformazione. Quando si utilizza un'applicazione AutoML, è necessario assicurarsi che le impostazioni predefinite utilizzate si adattino al business case e ai dati. Queste sono probabilmente sufficienti per problemi semplici e ben inquadrati ma potrebbero non essere sufficienti quando il business case richiede un approccio leggermente diverso o quando il set di dati non copre tutte le possibili situazioni in modo pulito. Quando si usa uno strumento low code per eseguire il training di un modello di Machine Learning, si avrà maggiore flessibilità. È possibile scegliere l'architettura, provare diversi valori per gli iperparametri, ottimizzarne uno o più e, in generale, eseguire esperimenti sui dati, l'architettura e gli iperparametri, prima di consegnare il modello all'applicazione di distribuzione. Gli strumenti low code rimuovono la barriera della programmazione, ma mantengono la libertà di sperimentare. Anche l'AutoML e gli strumenti low code non si escludono a vicenda, infatti è possibile creare una soluzione tramite entrambe le modalità.

Low-code machine learning

Le piattaforme di low-code machine learning sono simili alla loro controparte no-code, ma consentono comunque agli utenti di scrivere del codice. La percentuale di codice modificabile dipende dallo strumento. Analogamente alle piattaforme no-code, anche gli strumenti machine learning low-code sono utili per le aziende prive di professionisti specializzati AI. Gli strumenti di machine learning low-code aiutano a prevedere i tassi di abbandono, creare semplici modelli di riconoscimento di immagini, ottimizzare flussi di lavoro e creare sistemi di raccomandazione per diversi settori. Possono accelerare significativamente il processo di sviluppo del modello con set di impostazioni e dati già pronti. Ad esempio, AI Builder di Microsoft consente agli utenti di creare e gestire facilmente modelli di machine learning per elaborare il testo, prevedere i risultati aziendali e analizzare le opinioni dei clienti. Viso.ai è un'altra piattaforma utile per lo sviluppo di applicazioni di visione artificiale. Quindi, le piattaforme low-code consentono alle persone non esperte di trovare soluzioni a problemi semplici senza affidarsi a data scientist. Oltre a ridurre la dipendenza da essi, i dipendenti meno esperti hanno anche la possibilità di capire in che modo esattamente i dati influiscono sulle loro decisioni. Ad esempio, i "marketer" possono utilizzare tali strumenti per prevedere i tassi di abbandono o comprendere rapidamente l'attuale clima di mercato. Ciò consentirà loro di prendere decisioni basate sui dati e rimanere aggiornati. Gli esperti di marketing possono anche utilizzare strumenti di automazione low-code per configurare il chatbot di un sito Web con un approccio basato sull'elaborazione del linguaggio naturale (NLP). Ad esempio, lo strumento può aiutare a identificare le domande più frequenti e preparare il chatbot a prendere provvedimenti proattivi. Entro il 2030, le piattaforme di sviluppo low-code dovrebbero generare un fatturato di 187 miliardi. La crescita della piattaforma si basa sulla sua capacità di manipolare parte del codice (al contrario degli strumenti no-code), offrendo al low-code una migliore flessibilità di personalizzazione secondo i requisiti aziendali.

Capitolo 3

Architettura implementata

Come spiegato nel capitolo 2, l'obiettivo è quello di fornire dati e informazioni utili all'azienda, le quali saranno utilizzate dai decisori aziendali per definire e supportare le strategie di business, al fine di compiere le migliori scelte con l'obiettivo di trarre vantaggi competitivi, migliorare le prestazioni operative e la profittabilità e, più in generale, creare valore per l'azienda. Il progetto in questione permette all'azienda a rispondere in modo adeguato a molti di questi quesiti dei processi aziendali. Nel nostro caso si tratta di eseguire analisi predittive, più precisamente attraverso lo studio di una serie temporale. In particolare, il caso di studio tratta di un'azienda che vuole entrare nell'industria 4.0. L'azienda è caratterizzata da una fabbrica che produce bombole di gas. Essa possiede delle macchine che producono dei pezzi di bombole in una certa quantità; le bombole sono prodotte attraverso alcuni passaggi, tra cui ad esempio taglio, saldatura e collaudo. Ogni passaggio è caratterizzato da una macchina, la quale esegue l'azione e manda avanti il pezzo nella catena di montaggio. Su alcune di queste macchine è presente un conta pezzi che, ogni 10 minuti, ci indica il numero di pezzi prodotti da quella particolare macchina. Il nostro compito si incentra sul fare uno studio sull'andamento dei pezzi prodotti da una singola macchina, fare una previsione giornaliera per il futuro e fare dei ragionamenti what-if sulle relazioni fra i vari reparti di macchine. Ad esempio, se una macchina si guasta e smette di produrre, che andamento dovranno assumere le altre macchine del reparto per sopperire a questa mancanza. Da un punto di vista tecnologico, il programma dovrà, attraverso API prendere ogni tot periodo di tempo in input (ad esempio giornalmente), il valore di pezzi prodotti da ogni macchina, e continuare a mantenere un modello adeguato di predizione per i giorni seguenti. I risultati dell'algoritmo a loro volta saranno mandati attraverso api ad una applicazione sviluppata in low code (wavemaker). L'accesso ai dati dei pezzi prodotti dalle macchine avviene attraverso sdk di amazon AWS timestream, la quale legge i dati su serie storica in modo ottimale.

3.1 Architettura funzionale

L'architettura del progetto, come mostrato in figura 3.1, è composta da tre elementi principali.

- Una parte hardware costituita dalla parte fisica del sistema cioè' il dispositivo contatore di pezzi e il dispositivo ECU.
- Una parte di back-end caratterizzata dalla gestione dei dati ricevuti dal dispositivo conta pezzi della macchina sulla catena di montaggio e una parte di computazione di alcuni algoritmi predittivi. Questa parte è caratterizzata dalla creazione della parte di algoritmi di machine learning con successivo consumo e utilizzo di una certa quantità di dati. I dati riguardanti il numero totale di pezzi prodotti dalla macchina dell'industria, vengono rilevati e salvati a ritmo di dieci minuti, quindi la frequenza di dati è abbastanza alta. Alcuni sensori contano i pezzi poi attraverso il protocollo OPC UA i dati vengono trasferiti ad un'unità elettronica di controllo; infine, attraverso il protocollo MQTT, avviene il passaggio finale di trasferimento dall'unità fino ad un database, da cui, per mezzo di alcune query, i dati vengono scaricati su una piattaforma per lo storage. Da qui, attraverso alcune API i dati vengono analizzati e processati da alcuni algoritmi di machine learning, per arrivare infine ai risultati finali desiderati, anche essi caricati sullo storage, ed infine presi e visualizzati sulla parte front-end.
- Una parte front-end che permette di visualizzare i risultati dei dati analizzati precedentemente. Questa parte comprende l'applicazione web (o android) utilizzata per visualizzare a schermo i risultati dell'analisi e predizione eseguita dagli algoritmi di machine learning presenti nella parte back-end. L'applicazione comprende un login da eseguire per poter visualizzare i dati su un grafico temporale. I dati vengono scaricati dallo storage attraverso alcune API.

Di seguito si discutono i componenti principali delle parti hardware e di back-end.

Dispositivo contatore

Come primo passaggio, i pezzi vengono contati da un dispositivo installato sulla macchina sulla catena di montaggio. Il dispositivo incrementa un contatore ogni volta che passa un pezzo di bombola, una volta ogni dieci minuti il numero risultante viene registrato su un secondo dispositivo ed infine il contatore viene resettato a zero e si ricomincia.

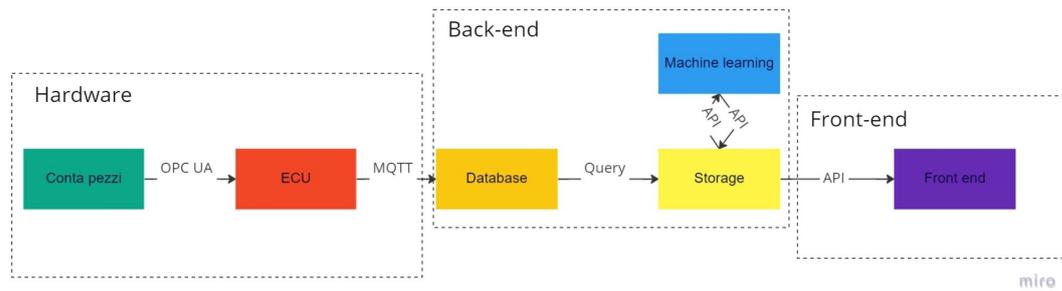


Figura 3.1: Architettura funzionale

OPC UA Protocol

La modalità con il quale in numero viene passato al secondo dispositivo è attraverso il protocollo OPC UA. [7] Open Platform Communications (OPC) e OPC UA (Unified Architecture) sono standard che facilitano lo scambio di dati tra controllori logici programmabili (PLC), interfacce uomo-macchina (HMI), server, client e altri macchinari ai fini dell'interconnettività e del passaggio di informazioni. L'OPC UA, rilasciata nel 2008, è un'architettura orientata ai servizi indipendente dalla piattaforma che integra tutte le funzionalità delle singole specifiche OPC Classic in un unico framework estensibile. Di seguito alcune importanti caratteristiche di OPC UA.

- **Equivalenza funzionale:** basandosi su OPC Classic, OPC UA è stato progettato per migliorare e superare le capacità delle specifiche OPC Classic.
- **Indipendenza di piattaforma:** OPC UA fornisce l'infrastruttura necessaria per l'interoperabilità in tutta l'azienda, da macchina a macchina, da macchina a azienda e tutto il resto.
- **Sicurezza:** OPC UA è compatibile con i firewall e risolve i problemi di sicurezza fornendo una suite di controlli.
- **Estensibilità:** tecnologie e metodologie innovative come nuovi protocolli di trasporto, algoritmi di sicurezza, standard di programmazione o servizi applicativi possono essere incorporate in OPC UA mantenendo la retro compatibilità per i prodotti esistenti.

ECU

Un'unità di controllo elettronico (ECU) è un piccolo dispositivo che è responsabile del controllo di una funzione specifica. Viene utilizzato molto nell'ambiente auto-

mobilitivo. Ogni ECU contiene in genere un chip dedicato che esegue il proprio software o firmware e richiede connessioni di alimentazione e dati per funzionare. Una centralina riceve input da diverse parti del veicolo, a seconda della sua funzione. Ad esempio, in questo caso, l'input è programmato in modo che arrivi una volta ogni dieci minuti.

MQTT Protocol

MQTT [9] (Message Queuing Telemetry Transport) è un protocollo di messaggistica sviluppato per ambienti vincolati come M2M (Machine to Machine) e IoT (Internet of Things). MQTT si basa sul principio di messaggistica Pub/Sub della pubblicazione di messaggi e sottoscrizione di argomenti. Il protocollo impacchetta in modo efficiente i messaggi per ridurre al minimo il sovraccarico.

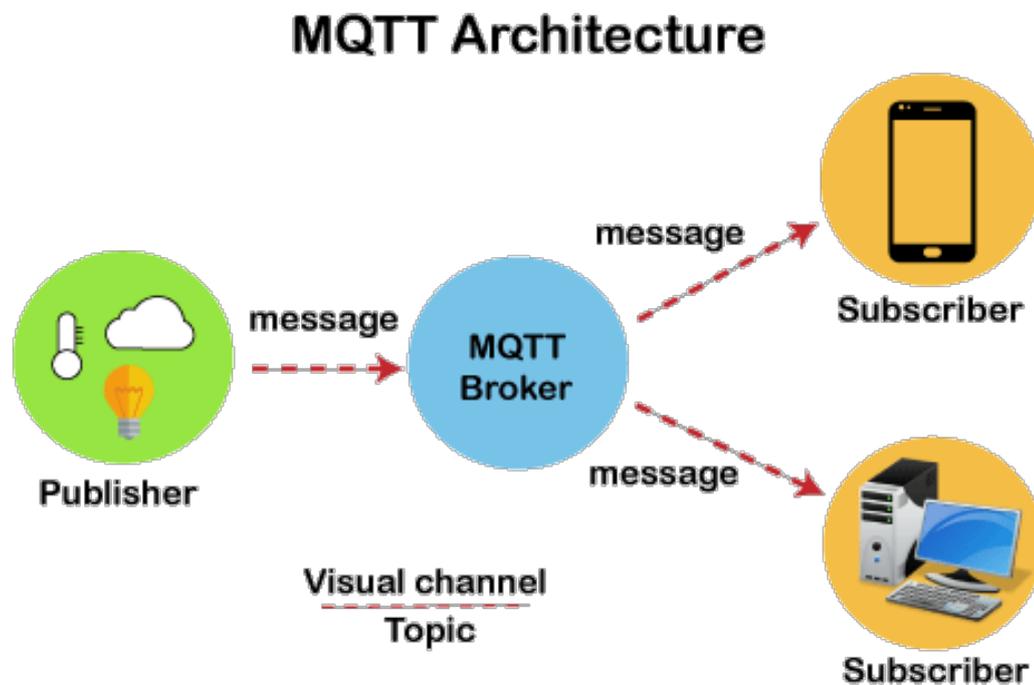


Figura 3.2: Esempio architettura mqtt

Database

Il database utilizzato è basato su serie temporale. [8] Un database di serie temporali (TSDB) è un database ottimizzato per i dati con timestamp o serie temporali.

I dati delle serie temporali sono semplicemente misurazioni o eventi che vengono tracciati, monitorati, sottoposti a down sampling e aggregati nel tempo. Potrebbe trattarsi di metriche del server, monitoraggio delle prestazioni delle applicazioni, dati di rete, dati dei sensori, eventi, clic, scambi in un mercato e molti altri tipi di dati analitici. Un database di serie temporali viene creato appositamente per la gestione di metriche ed eventi o misurazioni con timestamp. Un TSDB è ottimizzato per misurare il cambiamento nel tempo. Le proprietà che rendono i dati delle serie temporali molto diversi rispetto ad altri carichi di lavoro di dati sono la gestione del ciclo di vita dei dati, il riepilogo e le scansioni a intervalli estesi di molti record.

Storage e machine learning

Lo storage (in questo caso S3) è utilizzato per salvare e archiviare i dati che verranno inviati tramite API alla parte di machine learning (Libreria AWS Boto3); i dati salvati sullo storage sono sia di input, cioè i dati ricevuti dalle misurazioni dei dispositivi in fabbrica, sia di output, cioè, una volta analizzati dagli algoritmi di machine learning, sia avrà come risultato un file di dati con la struttura simile a quella di input, rappresentanti le predizioni per un dato periodo successivo. La componente machine learning è duale, infatti, per motivi di sperimentazione e testing, è stata realizzata sia tramite la creazione manuale di uno script python, sia attraverso la componente AWS Forecast, la quale sarà spiegata nelle prossime sezioni.

3.2 Architettura tecnologica

Per quanto riguarda il livello tecnologico si è deciso di utilizzare AWS tramite alcuni suoi servizi. AWS è una piattaforma in cloud che offre numerosi servizi a livello globale.

3.2.1 back-end

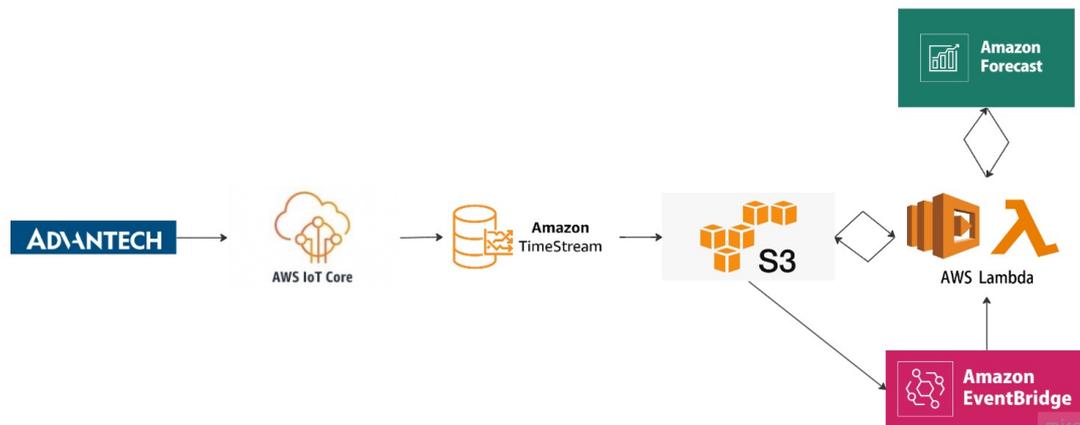


Figura 3.3: Architettura tecnologica back-end

Dispositivo Advantech

È il dispositivo ECU che ha come compito quello di gestire il conteggio di pezzi. L' ECU è fornito dall'azienda Advantech, la quale fornisce soluzioni hardware e software per il mondo industriale.

AWS IoT Core

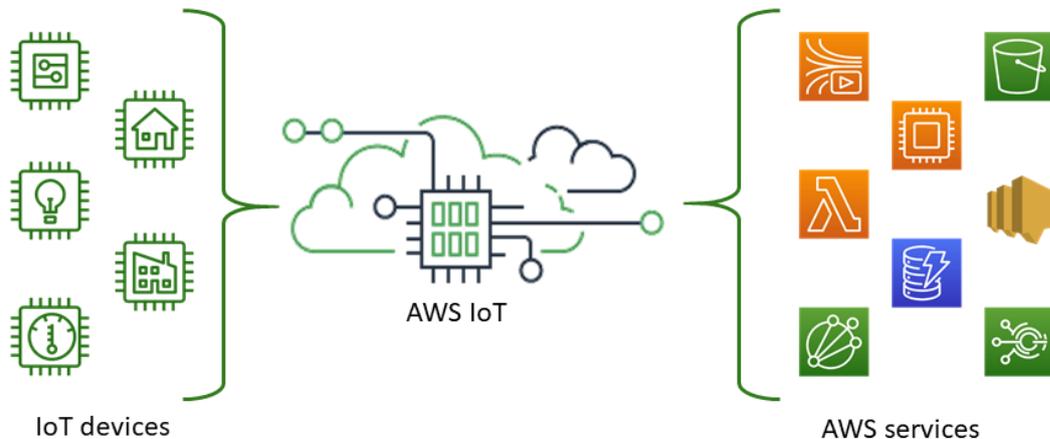


Figura 3.4: AWS IoT Core

AWS IoT core viene utilizzato per connettere dispositivi IoT ai vari servizi di AWS. In questo caso è stato utilizzato per connettere e trasferire i dati dal dispositivo advantech ad AWS Timestream. IoT Core fornisce diverse interfacce di connessione, ad esempio, nel progetto è stato utilizzato il protocollo Mqtt, già menzionato precedentemente.

AWS Lambda

AWS Lambda [5] è un servizio di elaborazione serverless fornito da Amazon Web Services (AWS). Attraverso AWS Lambda è possibile creare funzioni e applicazioni (scritte in uno dei linguaggi e runtime supportati) che vengono caricate in AWS Lambda, la quale esegue tali funzioni in modo efficiente e flessibile. Nel nostro caso, nel momento del caricamento di file su S3, viene notificato e parte la funzione lambda che utilizza il file appena caricato per eseguire una nuova predizione su AWS forecast. Le funzioni Lambda possono eseguire qualsiasi tipo di attività di elaborazione, dalla pubblicazione di pagine Web e flussi di dati alla chiamata di API e all'integrazione con altri servizi AWS. Il concetto di elaborazione "serverless" si riferisce al non dover mantenere i propri server per eseguire queste funzioni. AWS Lambda è un servizio completamente gestito che si occupa di tutta l'infrastruttura per l'utente. Quindi "serverless" significa solo che i server, i sistemi operativi, il livello di rete e il resto dell'infrastruttura sono già stati

curati, in modo da potersi concentrare sulla scrittura del codice dell'applicazione. Per quanto riguarda il funzionamento effettivo, ogni funzione Lambda viene eseguita nel proprio "contenitore". Quando viene creata una funzione, Lambda la impacchetta in un nuovo contenitore e quindi esegue tale contenitore su un cluster multi-tenant di macchine gestite da AWS. Prima che le funzioni inizino a funzionare, al contenitore di ogni funzione viene allocata la RAM necessaria e la capacità della CPU. Una volta terminata l'esecuzione delle funzioni, la RAM allocata all'inizio viene moltiplicata per la quantità di tempo trascorso dalla funzione in esecuzione. Gli utenti vengono quindi addebitati in base alla memoria allocata e alla quantità di tempo di esecuzione che la funzione ha impiegato per completare. Poiché il servizio è completamente gestito, l'utilizzo di AWS Lambda può far risparmiare tempo nelle attività operative. Quando non c'è un'infrastruttura da mantenere, si può dedicare più tempo a lavorare sul codice dell'applicazione, anche se ciò significa anche rinunciare alla flessibilità di gestire l'infrastruttura. Una delle proprietà architettoniche distintive di AWS Lambda è che molte istanze della stessa funzione o di funzioni diverse dello stesso account AWS possono essere eseguite contemporaneamente.

AWS Timestream

AWS Timestream è un servizio serverless che offre un database ottimizzato per la gestione di serie temporali. Esso semplifica l'archiviazione e l'analisi dei dati su serie temporale. Inoltre, il motore di query permette di accedere e analizzare i dati recenti e storici insieme, senza dover specificare nella query se i dati risiedono a livello di memoria o di ottimizzazione dei costi. Timestream è serverless e si dimensiona automaticamente per adattarsi alla capacità e alle prestazioni specifiche. Questo elimina la necessità di gestire l'infrastruttura sottostante e permette di creare liberamente le applicazioni.

AWS S3

AWS S3 [6] è un servizio che offre uno spazio di memoria in cloud utilizzato per memorizzare e archiviare i file caricati. Viene usato per fare upload del file csv dei dati contenenti le informazioni riguardo i pezzi prodotti dalle macchine in fabbrica. I dati sono archiviati come oggetti all'interno di risorse chiamate "bucket"; Un oggetto è un file e tutti i metadati che descrivono il file. Un bucket è un contenitore per oggetti. Per archiviare i dati in Amazon S3, bisogna prima creare un bucket e specificare un nome e una regione AWS. Quindi, è possibile caricare i dati in quel bucket come oggetti. Ogni oggetto ha una chiave, che è l'identificatore univoco per l'oggetto all'interno del bucket. S3 fornisce diverse funzionalità possibili da configurare per supportare il proprio caso d'uso specifico.

Ad esempio, è possibile utilizzare S3 Versioning per mantenere più versioni di un oggetto nello stesso bucket, che consente di ripristinare gli oggetti eliminati o sovrascritti accidentalmente. I bucket e gli oggetti in essi contenuti sono privati e sono accessibili solo se si concedono esplicitamente le autorizzazioni di accesso. È possibile utilizzare le policy dei bucket, le policy di AWS Identity and Access Management (IAM), gli elenchi di controllo degli accessi (ACL) e gli Access Point S3 per gestire l'accesso. Si può anche impostare una gestione di autorizzazione agli accessi di ogni singolo bucket o file, in modo da permettere solo ad alcuni ruoli di accedervi. S3 include anche altre funzionalità, come ad esempio aggiungere tag di metadati agli oggetti, spostare e archiviare i dati tra le classi di archiviazione, proteggere i dati da utenti non autorizzati, monitorare i dati a livello di oggetto e di bucket e visualizzare l'utilizzo dell'archiviazione.

AWS EventBridge

AWS EventBridge [3] è un router di eventi serverless che permette di ricevere, filtrare, trasformare, instradare e consegnare eventi. Può essere utilizzato per connettere applicazioni ai dati provenienti da varie origini. EventBridge riceve un evento, un indicatore di un cambiamento nell'ambiente e applica una regola per indirizzare l'evento a un bersaglio. Le regole abbinano gli eventi agli obiettivi in base alla struttura dell'evento, chiama un modello di eventi in base a un programma. Ad esempio, quando viene caricato un file su S3, si può avere una regola che invia l'evento a una funzione Lambda, mandandola in esecuzione. Quindi permette di coordinare in modo ottimale eventi presenti in AWS.

AWS Forecast

Amazon Forecast [4] è un servizio che utilizza algoritmi statistici e di apprendimento automatico per fornire previsioni di serie temporali. Basato sulla stessa tecnologia utilizzata per la previsione di serie temporali in Amazon, Forecast fornisce algoritmi vari per prevedere i dati delle serie temporali future in base ai dati storici e non richiede esperienza di apprendimento automatico. La previsione delle serie temporali è utile in diversi campi tra cui finanza, logistica e medicina.

3.2.2 Front-end

Wavemaker

Il programma si presenta con un'interfaccia user friendly, con la possibilità di aggiungere facilmente diversi elementi grafici, tra cui form, liste, bottoni e slider. Questi elementi possono anche essere modificati graficamente attraverso scrittura

di css o tramite interfaccia. Ad ogni bottone è possibile associare una diversa variabile da eseguire in un determinato momento, come ad esempio sulla pressione. Una variabile rappresenta un "oggetto" con il quale è possibile passare informazioni e compiere diverse azioni. Ogni azione può agire su un determinato ambito dell'applicazione, come ad esempio database, api e dispositivo. Molti elementi dell'applicazioni possono essere scritti anche attraverso codice, ad esempio aggiungendo elementi tramite html oppure azioni da svolgere attraverso javascript. Su wavemaker è possibile collegare un qualsiasi database per poi fare query sql su esso. è inoltre possibile aggiungere le proprie API o utilizzarle per fare operazioni sul database.

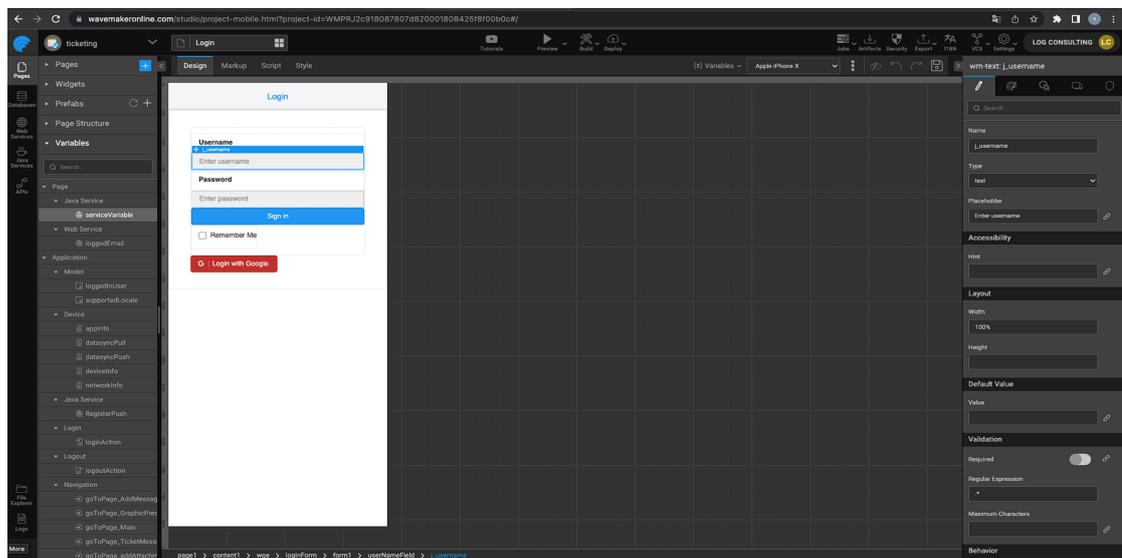


Figura 3.5: Schermata raffigurante l'interfaccia di wavemaker

3.3 Implementazione

Modellazione dei dati

Il database è caratterizzato dalla presenza di alcuni file CSV. Ogni riga di csv contiene un record della tabella. I record possiedono cinque campi:

- timestamp dell'avvenuta misurazione;
- nome dell'organizzazione e azienda;
- nome della macchina presente in fabbrica;

- dipartimento di lavorazione;
- numero di pezzi prodotti da quella macchina in quel determinato momento.

Una volta importati i dati sullo script python, dopo aver filtrato le macchine che producono un numero di pezzi diverso da zero (quindi si controlla che almeno funzionino), viene scelta una macchina. Infine, è possibile raggruppare i pezzi prodotti o per giorno intero, quindi avere meno dati da analizzare, oppure mantenere la mole di dati di default, caratterizzata da pezzi prodotti ad un intervallo di dieci minuti. L'analisi in questione è stata compiuta considerando il raggruppamento dei dati giornaliero, ma non si esclude come studio futuro l'analisi per intervalli di dieci minuti. Alla fine, di queste operazioni di filtraggio e raggruppamento, i dati finali che serviranno da input ai modelli predittivi sarà una lista contenente la serie temporale nel seguente formato:

- indice rappresentante la dimensione temporale: partendo da zero e in ordine crescente, ogni singolo giorno all'interno della sequenza temporale è rappresentata da un indice.
- valore totale: intero rappresentante il numero di pezzi prodotti in quel determinato giorno da quella determinata macchina.

Il formato di output, come sarà anche mostrato successivamente nel metodo "moving window", è uguale a quello di input, l'unica differenza è che vengono mostrati valori su periodi di giorni successivi a quelli di input.

3.3.1 Modelli predittivi

La componente machine learning è servita per poter fare predizioni su serie storica. Sono stati utilizzati due algoritmi principali: una rete neurale MLP e un algoritmo basato su metodi statistici, cioè SARIMA. Tutto questo è stato implementato attraverso la scrittura di codice Python su un notebook Jupyter. Per quanto riguarda l'algoritmo di funzionamento della predizione attraverso la modalità rete neurale è attraverso "moving window". Il modello, dopo essersi allenato sul train set e valutato sul validation set, vengono presi in considerazione gli ultimi N valori predetti presenti sulla parte del validation set; N quindi corrisponde alla dimensione della finestra. A partire da questi N valori, il modello predice il prossimo valore; successivamente questo viene aggiunto in terza posizione, mentre il primo viene scartato (come in una coda FIFO); si va avanti così fino a quando vengono predetti la quantità di valori futuri desiderati. Quindi essendo i valori predetti dal modello (e non reali quindi), più si va avanti nel tempo, più i risultati potrebbero essere imprecisi rispetto alla realtà.

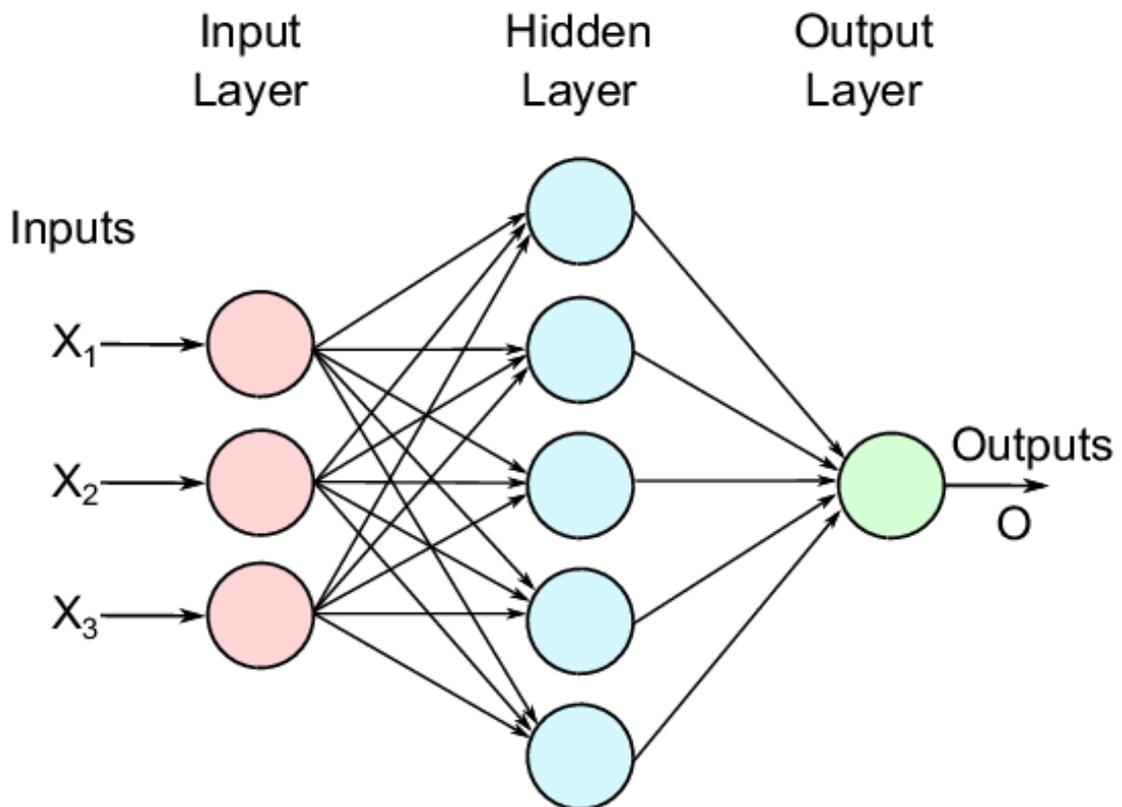


Figura 3.6: Esempio di una semplice rete neurale MLP

Metodo basato su rete neurale

Una rete neurale artificiale [10] è un sistema di processamento dati modellato sul funzionamento dei neuroni nel cervello umano. Le reti neurali artificiali costituiscono una varietà di tecnologie di deep learning, le quali rientrano anche nell'intelligenza artificiale o AI. È uno dei metodi più recenti e performanti per risolvere problemi moderni che vanno da una grande varietà di campi, dall'informatica alla medicina e tecnologia. In questo progetto si è deciso di utilizzare una rete neurale completamente connessa chiamata Multilayer Perceptron (MLP), raffigurata in figura 3.6. Ha tre livelli di neuroni di cui uno nascosto. Se ha più di un livello nascosto, viene chiamata ANN profonda (Deep Artificial Neural Network). MLP è un tipico esempio di rete neurale artificiale "feedforward". Con "feedforward" si intende una rete neurale artificiale le cui connessioni non formano cicli (a differenza della rete neurale ricorrente). In generale il numero di strati e il numero di neuroni di una rete neurale possono essere molto variabili e rientra nella bravura del programmatore riuscire a trovare i valori più adatti. Esistono tecniche

che aiutano a trovare questi valori ideali, come ad esempio la "cross validation" oppure "grid search". La procedura di regolazione dei pesi dei neuroni [1] viene eseguito tramite backpropagation. Le reti neurali più profonde sono migliori nell'elaborazione dei dati, tuttavia, strati più profondi possono portare a problemi di "vanishing gradient"; esistono particolari algoritmi per risolvere questi tipi di problemi. Per quanto riguarda gli iperparameteri del modello scelto, sono stati decisi attraverso alcuni test empirici attraverso vari tentativi di train e testing. Alla fine, si è optato per una rete neurale profonda MLP con gli strati nascosti contenenti cento e otto neuroni. Per quanto riguarda la funzione di attivazione si è scelta Relu perché rappresenta la soluzione più semplice per ridurre al minimo il problema del "vanishing gradient" dal momento che si satura soltanto in una direzione, essendo che i valori sotto lo zero vengono fissati a zero.

Metodo statistico

Autoregressive Integrated Moving Average [2], o ARIMA, è uno dei metodi di previsione più utilizzati per la predizione di dati di serie temporali univariate. Come suggerisce il nome, supporta sia elementi autoregressivi sia il "moving average". Il termine "integrato" si riferisce alla differenziazione che consente al metodo di supportare dati di serie temporali che possiedono un trend.

Gli aspetti chiave del modello sono:

- Autoregressione (AR): il modello utilizza una relazione tra un'osservazione e un certo numero di osservazioni "ritardate".
- Integrato (I): il modello utilizza la differenziazione delle osservazioni grezze (ad esempio, sottraendo un'osservazione da un'osservazione al passo temporale precedente) per rendere stazionaria la serie temporale.
- Media mobile (MA): il modello utilizza la dipendenza tra un'osservazione e un errore residuo da una media mobile di un modello applicato alle osservazioni ritardate.

Ciascuno dei suoi componenti è specificato in modo esplicito nel modello come parametro ARIMA(p,d,q), definito come segue:

- P: il numero di osservazioni di ritardo incluse nel modello, anche chiamato l'ordine di ritardo.
- D: il numero di volte in cui le osservazioni grezze sono differenziate, chiamato anche il grado di differenza.
- Q: la dimensione della finestra della media mobile, chiamata anche ordine di media mobile.

Un problema con ARIMA è che non supporta i dati stagionali. ARIMA si aspetta dati che non siano stagionali o che abbiano la componente stagionale rimossa, ad esempio destagionalizzati tramite metodi come la differenziazione stagionale. Un'alternativa è usare SARIMA. Seasonal Autoregressive Integrated Moving Average, SARIMA o Seasonal ARIMA, è un'estensione di ARIMA che supporta esplicitamente dati di serie temporali univariate con una componente stagionale. Aggiunge tre nuovi iperparametri per specificare l'autoregressione, la differenziazione e la media mobile per la componente stagionale della serie, nonché un parametro aggiuntivo per il periodo della stagionalità. Notazionalmente, sono indicati con $ARIMA(p,d,q)(P,D,Q)_m$ dove:

- p,d,q sono i parametri ARIMA riferiti ai periodi.
- P,D,Q gli stessi parametri riferiti alle stagioni.
- M : numero di periodi in una stagione (ad esempio settimane in un trimestre).

3.3.2 Front-end

Per poter utilizzare le API di AWS e per poter accedere ai risultati delle predizioni temporali, si è reso necessario utilizzare la libreria AWS SDK. Grazie a questo, è stato possibile accedere a S3, selezionare il bucket desiderato e scaricare il file contenente i dati richiesti.

Prima di tutto, è stata implementata una classe interna per poter gestire l'output dei dati in formato json. I dati da gestire erano tre: il nome dell'item, il timestamp con il suo valore corrispondente. La creazione del Client S3 è stata effettuata attraverso l'inserimento delle credenziali appartenenti all'account AWS e della regione di appartenenza (Europa). Successivamente, dopo aver scritto il nome del bucket e dell'oggetto, si procede con la connessione e download dell'item. Dopo aver ricavato il file, si legge riga per riga e si inseriscono i dati in una lista. Da qui la lista verrà letta e i dati ricavati inseriti e visualizzati a schermo in un grafico su serie temporale.

Capitolo 4

Test

Alcuni dei test principali riguardano un confronto fra le modalità di predizione tramite rete neurale e tramite metodo statistico Sarima. I test riguardano la precisione degli algoritmi predittivi. Si è data priorità alla precisione sulla velocità effettiva dell'esecuzione degli algoritmi perché quello che ci interessava era avere una qualità maggiore delle predizioni ricevute. Come accennato nella sezione precedente, si sono presi in esame dei dati raggruppati giornalmente, con la presenza di 81 record rappresentanti i giorni totali per la fase di allenamento e 10 per la fase validazione. Questa scelta è stata compiuta attraverso l'utilizzo dell'MSE come parametro di valutazione: infatti, controllando i risultati di vari split, da 60% a 90%, quello più basso (e quindi migliore) era quest'ultimo. Quindi, la misurazione della bontà dei risultati è avvenuta tramite l'utilizzo di "mean squared error" (MSE), che misura la media dei quadrati degli errori, ovvero la differenza quadratica media tra i valori stimati e il valore effettivo. I test sono stati eseguiti su due macchine presenti in fabbrica e si è avuto come risultato un MSE medio di circa 650. Di seguito due grafici per ogni macchina rappresentata: il primo mostra come si è comportato su train e validation, il secondo mostra la predizione eseguita per un certo numero di periodi successivi; quindi, su dati non presenti nel dataset iniziale.

Come si può vedere in figura 4.1, viene rappresentato il grafico dei risultati sul train e test set riguardo la macchina "CL05-RR" adibita al collaudo dei pezzi; il grafico blu rappresenta i valori storici reali, quello arancione la prestazione del modello sul train set, mentre quello verde la prestazione sul test set. Sull'asse delle ascisse sono segnati gli indici temporali rappresentanti i giorni, mentre sull'asse delle ordinate i valori totali dei pezzi prodotti. Si può notare come i risultati siano abbastanza aderenti a quelli reali, quindi un risultato abbastanza soddisfacente.

In figura 4.2, viene invece mostrata una predizione su 21 giorni successivi (tre settimane). Si può vedere invece, anche ad occhio nudo, come i risultati non sembrano molto corretti e veritieri; questo a causa della dimensione abbastanza

ridotta della quantità di dati a disposizione.

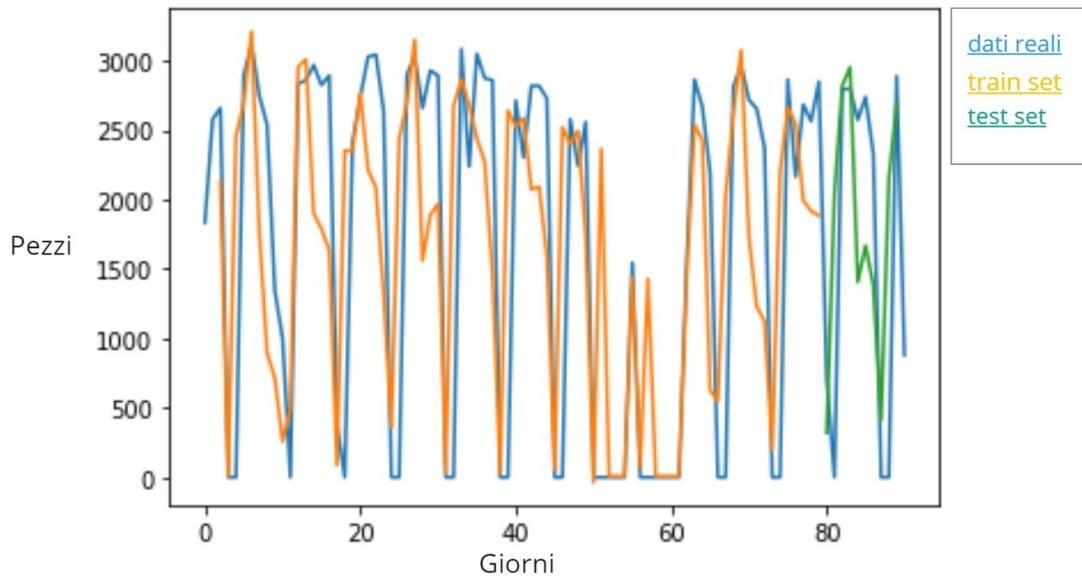


Figura 4.1: Grafico della macchina 1 con corrispondenza fra train e test set

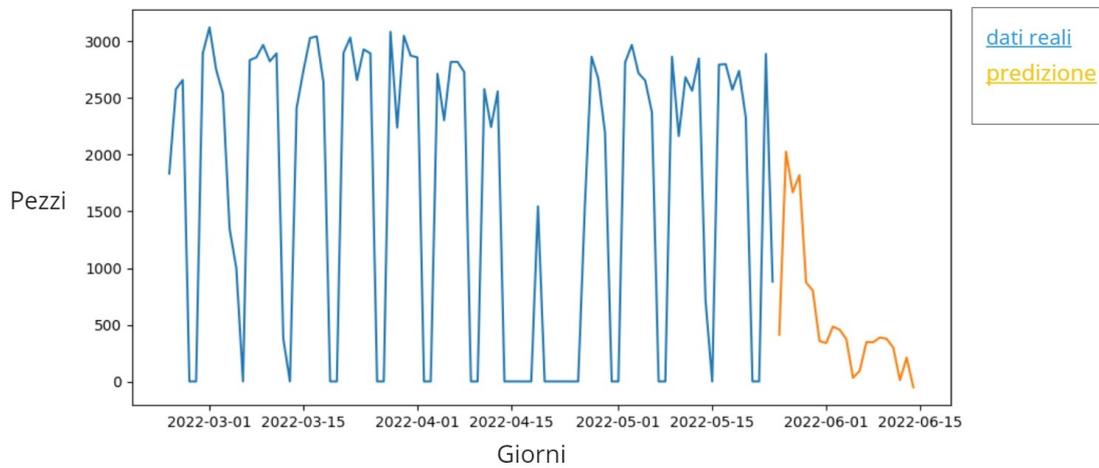


Figura 4.2: Grafico della macchina 1 mostrante la predizione su N periodi successivi

In figura 4.3, si può notare come il grafico rappresenti una certa ciclicità, infatti è presente una stagionalità settimanale: si possono vedere periodi rappresentati

da cinque "crescite", separati da periodi dove il valore è a zero; questo significa che le macchine lavoravano da lunedì a venerdì, con le sole pause durante la notte e durante i weekend.

La figura 4.4 rappresenta la stessa cosa con l'aggiunta della possibile predizione su tre settimane successive. Anche qui si può applicare il discorso della poca precisione dei risultati.

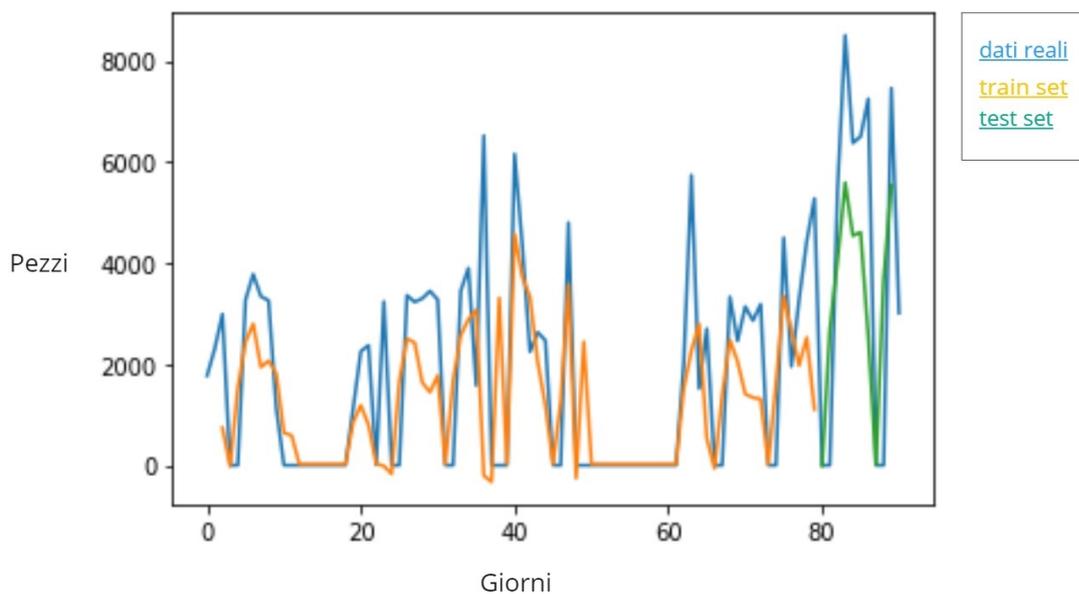


Figura 4.3: Grafico della macchina 2 con corrispondenza fra train e test set

Per quanto riguarda il metodo statistico con Sarimax, come si può vedere simmetricamente rispettivamente per la macchina 1 e 2, in figura 4.5 e in figura 4.6, si è avuto come risultato un MSE medio di circa 500. Quindi un risultato migliore, come descritto precedentemente. Anche i grafici risultano più aderenti e realistici ad una situazione più veritiera. Per quanto riguarda la legenda, i colori dei grafici rispecchiano i significati precedenti: blu per i dati reali, arancione per le predizioni sul train set, verde per le predizioni sul test set.

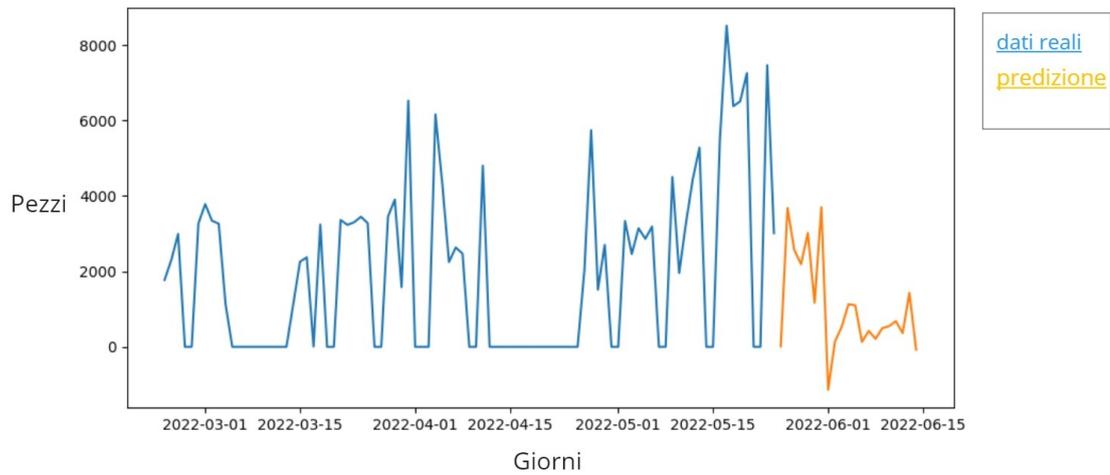


Figura 4.4: Grafico della macchina 2 mostrante la predizione su N periodi successivi

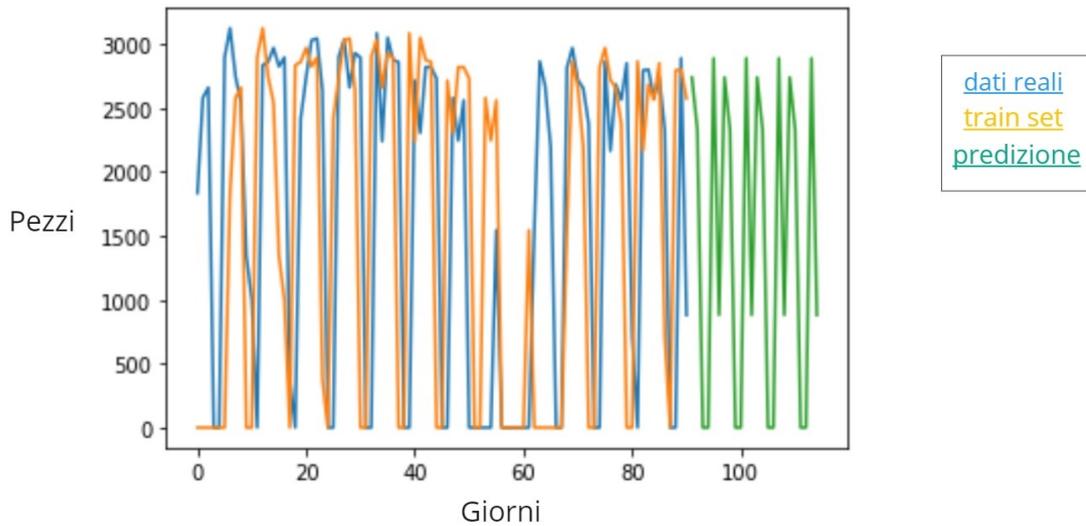


Figura 4.5: Grafico della macchina 1 mostrante la predizione effettuata con Sari-max

Siccome i risultati delle predizioni tramite metodo statistico hanno avuto risultati migliori, e considerando il fatto che la quantità di istanze nel dataset non era molto elevata, si può concludere che, a seguito anche della consultazione di

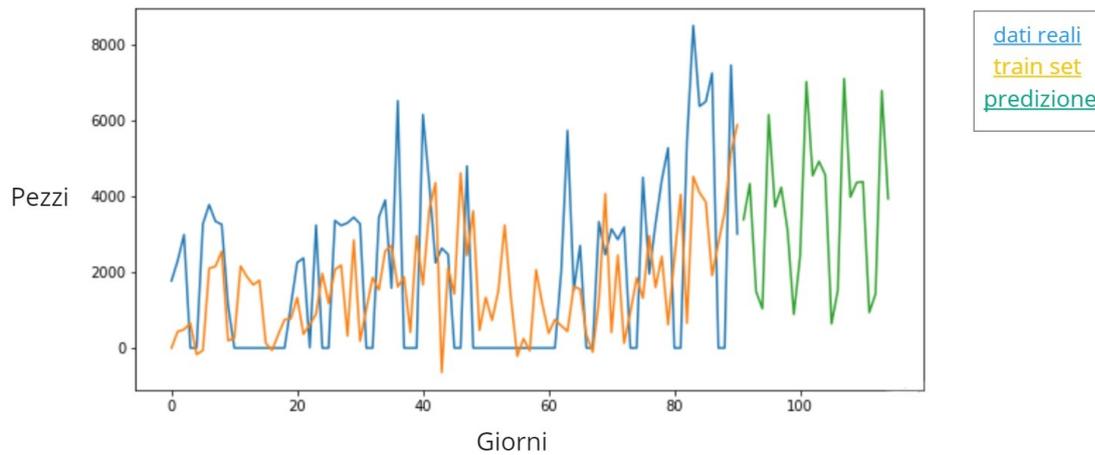


Figura 4.6: Grafico della macchina 2 mostrante la predizione effettuata con Sari-max

[11], in generale la rete neurale costituisce un potenziale maggiore riguardo alle prestazioni dei risultati, soprattutto quando si hanno a disposizione grandi quantità di dati, oltre ad una buona scelta di parametri; al contrario di SARIMA che restituisce risultati abbastanza soddisfacenti anche con pochi dati.

Per quanto riguarda il funzionamento generale dell'applicazione, la parte frontend funziona come desiderato e, anche se non sono stati fatti particolari test di velocità, i risultati e i dati delle predizioni avvengono in tempi istantanei. I risultati vengono mostrati in modo corretto.

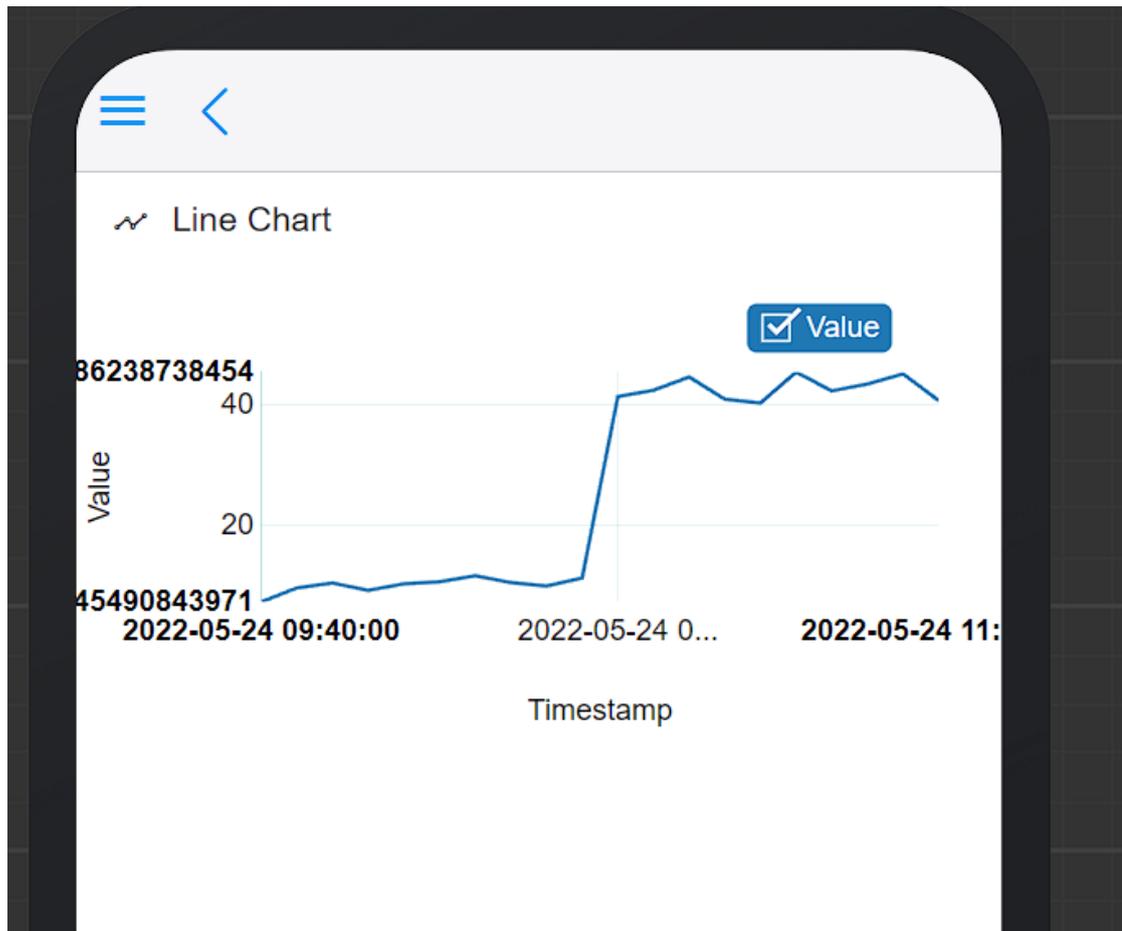


Figura 4.7: Risultati mostrati sull'applicazione creata con Wavemaker

In figura 4.7 è possibile vedere il risultato finale delle predizioni effettuate tramite i metodi precedenti; dopo aver effettuato la predizione, i risultati vengono messi in un file CSV sullo spazio di archiviazione di AWS S3; infine, tramite il collegamento API, l'applicazione creata su Wavemaker, prende i risultati e vengono mostrati a schermo come nell'immagine.

Capitolo 5

Conclusioni

L'obiettivo della tesi era quello di fornire degli strumenti che aiutassero l'azienda ad entrare nell'industria 4.0. Nello specifico creare un'applicazione utilizzando tecnologie moderne, come ad esempio tecniche di machine learning e low-code, per fornire delle analisi predittive ad un'azienda riguardanti i pezzi prodotti in fabbrica da un determinato macchinario. Si tiene però a precisare che l'obiettivo della tesi non era tanto raggiungere dei risultati predittivi migliori in assoluto, ma sperimentare e utilizzare le nuove tecnologie spiegate nei capitoli precedenti. Nello sviluppo della tesi è stato prima di tutto necessario approfondire l'argomento industria 4.0 e low code per comprendere meglio queste tecnologie e, di conseguenza, utilizzarle al meglio. Da ciò è stato possibile concentrarsi su degli obiettivi più specifici: lo strumento doveva essere in grado, data una serie storica di una macchina produttrice di pezzi, di analizzare l'andamento e predire la quantità di pezzi prodotti per N periodi successivi, ad esempio per le prossime due settimane. Le analisi sono state anche di tipo what-if, cioè all'avvenuta di una modifica nella catena di produzione (ad esempio un rallentamento causato da un guasto), che tipo di andamento produttivo dovranno assumere le altre macchine? Queste informazioni potrebbero essere sfruttate dall'azienda in vari modi, per fare le scelte giuste in ambito di business intelligence, ad esempio modificando adeguatamente il ritmo del flusso di produzione in base alle richieste del mercato. Per raggiungere tale obiettivo si è progettata e implementata un'architettura in cloud basata sui servizi offerti da AWS in grado di raccogliere, estrarre e salvare i dati in un sistema di archiviazione, chiamato S3. Poi in un secondo momento è stata realizzata, tramite la piattaforma low-code Wavemaker, la Dashboard, cioè la parte di interfaccia visiva che si occupa di far partire le varie interrogazioni e analisi sul database e restituire i risultati in una tabella o grafico e rappresentarli graficamente. Dalla Dashboard (o parte front-end) è possibile visualizzare in modo intuitivo tutta la rete di interazioni con le relative statistiche e i risultati delle analisi, raggiungendo così l'obiettivo prestabilito. Gli obiettivi che ci si è posti rappresentano solo una

parte di ciò che è possibile realizzare con queste tecnologie. Durante lo sviluppo sono emersi anche altri aspetti da approfondire:

- il primo aspetto è sicuramente rappresentato dalla possibilità di fare ulteriori approfondimenti in generale sulla parte tecnologica delle predizioni su serie storica;
- utilizzo di una serie storica basata sulla granularità di dieci minuti anziché giornaliera;
- come si comporterebbe il sistema avendo a disposizione una quantità maggiore di dati dal punto di vista della "densità", ad esempio anziché avere i dati ricevuti dai dispositivi in fabbrica ogni dieci minuti, averli ogni secondo;
- Eseguire un confronto più preciso fra le predizioni eseguite tramite metodo manuale in script python e metodo automatico tramite amazon forecast;
- Analizzare e testare altri metodi di machine learning per la predizione, come ad esempio le reti neurali ricorrenti.

Bibliografia

- [1] Massimo Buscema. Back propagation neural networks. *Substance use misuse*, 33:233–70, 02 1998.
- [2] Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj Moussami, and Abdeslam Lachhab. Forecasting of demand using arima model. *International Journal of Engineering Business Management*, 10:184797901880867, 10 2018.
- [3] <https://aws.amazon.com/it/eventbridge/>. Aws event bridge.
- [4] <https://aws.amazon.com/it/forecast/>. Aws forecast.
- [5] <https://aws.amazon.com/it/lambda/>. Aws lambda.
- [6] <https://aws.amazon.com/s3/>. Aws s3.
- [7] <https://opcfoundation.org>. Opc ua.
- [8] <https://www.influxdata.com>. Time series database.
- [9] <https://www.wolfssl.com>. Mqtt.
- [10] Marius-Constantin Popescu, Valentina Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8, 07 2009.
- [11] Patricia M. West, Patrick L. Brockett, and Linda L. Golden. A comparative analysis of neural networks and statistical methods for predicting consumer choice. *Marketing Science*, 16(4):370–391, 1997.