POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Modelli per la Gestione delle Scorte di Pezzi di Ricambio

Relatori
Prof. Brandimarte Paolo

Candidato
Montrucchio Matteo

Anno Accademico 2022-2023

Alla mia famiglia

Indice

C'è una meta
per il vento dell'inverno:
il rumore del mare.

[Ikenishi Gonsui]

Introduzione

L'evoluzione tecnologica ha rivoluzionato profondamente il modo in cui le aziende gestiscono le proprie operazioni e analizzano le prestazioni. In particolare, l'adozione diffusa dell'architettura cloud ha aperto nuove opportunità per l'ottimizzazione dei processi aziendali e la creazione di strumenti avanzati per l'analisi dei dati. Questa tesi si concentra sull'applicazione dell'architettura cloud per sviluppare un'applicazione web dedicata all'analisi e al miglioramento delle performance dei magazzini di pezzi di ricambio.

Nell'attuale panorama aziendale, la gestione efficiente dei magazzini di pezzi di ricambio è cruciale per garantire la continuità delle operazioni e la soddisfazione del cliente. La gestione dei pezzi di ricambio svolge un ruolo fondamentale nell'industria manifatturiera, nell'assistenza post-vendita e in una vasta gamma di settori. L'ottimizzazione di questi magazzini può comportare notevoli vantaggi in termini di costi, tempi di consegna e qualità del servizio. Al cuore di questa gestione delle scorte c'è la figura dell'Inventory Manager, un professionista con competenze specializzate nella pianificazione, nell'acquisto, nel controllo e nell'ottimizzazione degli inventari. L'Inventory Manager si trova a navigare in un ambiente aziendale sempre più complesso e dinamico, dove il margine di errore è ridotto al minimo. Le sue responsabilità includono la gestione delle previsioni della domanda, l'ottimizzazione dei punti di riordino, la negoziazione con i fornitori e il monitoraggio degli indicatori chiave di performance (KPI) di inventario. L'applicazione dell'architettura cloud rappresenta un passo avanti nella modernizzazione della gestione delle scorte. L'accesso a risorse scalabili e flessibili consente all'Inventory Manager di implementare soluzioni avanzate per affrontare le sfide della gestione degli inventari in modo più efficiente ed efficace. L'analisi dei dati in tempo reale e le capacità di elaborazione avanzate consentono di prendere decisioni più informate e reattive. Nel primo capitolo di questa tesi, verrà presentata l'infrastruttura cloud che è stata implementata, esploreremo come le opportunità offerte da questa nuova applicazione web siano state sviluppate e integrate, e come queste possano essere sfruttate dall'Inventory Manager per ottimizzare i processi di gestione delle scorte e migliorare le performance aziendali.

Nel secondo capitolo, si analizzerà nel dettaglio uno dei task fondamentali, la previsione della domanda. La domanda può variare notevolmente in base a diversi fattori, rendendo cruciale l'utilizzo di modelli di previsione adeguati e l'identificazione dell'algoritmo più adatto in diverse condizioni di domanda. Nell'ambito dell'applicazione web sviluppata, sono stati implementati diversi modelli di previsione della domanda. Questi modelli includono approcci statistici, machine learning e

tecniche avanzate di analisi dei dati. L'obiettivo è stato quello di confrontare l'accuratezza di questi modelli in diverse situazioni di domanda. La sfida principale è stata identificare quando un algoritmo di previsione offre risultati migliori rispetto agli altri. Questo problema è stato affrontato attraverso un processo di valutazione comparativa dei modelli, utilizzando metriche di performance appropriate. L'analisi dei modelli e la selezione dell'algoritmo ottimale sono aspetti fondamentali nell'applicazione web, poiché influenzano direttamente la precisione delle previsioni e la gestione delle scorte. Comprendere quale modello funzioni meglio in determinate situazioni di domanda permette all'azienda di prendere decisioni informate e ottimizzare ulteriormente i processi.

Infine, nel terzo capitolo, verrà presentato come calcolare diversi KPI per la gestione del magazzino, come punto di riordino e scorta di sicurezza, e alcuni indicatori fondamentali per la valutazione delle prestazioni dei magazzini, come il livello di servizio e il fill rate. Il livello di servizio misura l'efficienza nella soddisfazione delle richieste dei clienti, mentre il fill rate riflette la capacità del magazzino di completare gli ordini in modo completo e tempestivo. Inoltre, nel capitolo finale si esaminerà approfonditamente gli aspetti logistici legati alla distribuzione e all'organizzazione dei magazzini. La gestione logistica svolge un ruolo cruciale nella catena di approvvigionamento e nell'efficacia complessiva dei magazzini. Analizzeremo come questi aspetti possano essere sfruttati nella gestione delle scorte. Questi aspetti logistici rappresentano una parte integrante della valutazione delle performance dei magazzini di pezzi di ricambio. La comprensione e l'analisi dei KPI insieme agli aspetti logistici ci consentiranno di valutare in modo completo l'efficacia dei processi aziendali, individuare aree di miglioramento e prendere decisioni strategiche basate su dati concreti.

1 L'infrastruttura

1.1 Cloud computing

L'ambito del cloud computing è caratterizzato da una molteplicità di definizioni, ognuna delle quali offre una prospettiva unica su questo campo. In termini essenziali, il cloud computing può essere compreso come l'utilizzo di una vasta gamma di servizi, tra cui piattaforme di sviluppo software, server, spazio di archiviazione e applicazioni, tramite la rete. In generale, il termine "cloud computing" fa riferimento all'erogazione di servizi attraverso la rete, concetto ampiamente riconosciuto e accettato. Esso rappresenta una forma di esternalizzazione dei programmi informatici, consentendo agli utenti di accedere a software e applicazioni da qualsiasi luogo. Queste applicazioni sono gestite da terze parti e risiedono all'interno del cloud, permettendo agli utenti di liberarsi da preoccupazioni legate a dettagli tecnici come lo spazio di archiviazione e la potenza di calcolo, concentrandosi invece sull'utilizzo delle soluzioni finali. Invece di conservare i dati su un disco rigido o un dispositivo di archiviazione locale, l'archiviazione basata su cloud consente di memorizzare informazioni in un database remoto. L'accesso ai dati e ai programmi software è possibile ovunque si disponga di una connessione a Internet tramite un dispositivo elettronico. In commercio esistono diverse piattaforme cloud, tra le più diffuse troviamo Microsoft Azure, Amazon Web Services (AWS) e Google Cloud Platform (GCP), ciascuna con le sue caratteristiche distintive.

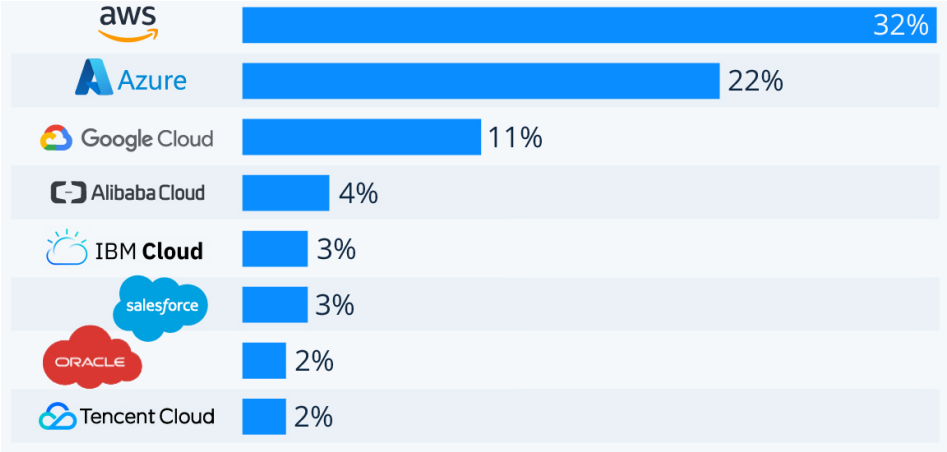

Figura 1.1: Una panoramica delle piattaforme cloud più diffuse.

Il termine "cloud" deriva dal simbolo a forma di nuvola comunemente utilizzato per rappresentare Internet in diagrammi e schemi. Inoltre, il nome "cloud computing" riflette il fatto che le informazioni accessibili si trovano in un luogo remoto all'interno di uno spazio virtuale. Le aziende che offrono servizi basati su cloud consentono agli utenti di archiviare file e applicazioni su server remoti, consentendo loro di accedere a tutte le informazioni tramite Internet, indipendentemente dalla loro posizione fisica. Ciò significa che gli utenti non sono vincolati a una posizione specifica per accedere ai dati, aprendo la possibilità di lavorare in modalità remota. Alcuni ritengono che il termine "cloud computing" sia stato inflazionato eccessivamente dalle grandi aziende software attraverso i loro reparti di marketing. Una critica comune è che questa tecnologia potrebbe portare all'abbassamento del livello di controllo delle organizzazioni sui propri dati, ad esempio quando un provider di servizi e-mail memorizza dati in diverse località in tutto il mondo. Grandi aziende, come le banche, potrebbero richiedere di mantenere i dati entro i confini nazionali. Sebbene non rappresenti un problema insuperabile, questa situazione evidenzia le possibili sfide che alcune organizzazioni potrebbero incontrare nel contesto del cloud computing.

Di seguito, esamineremo il concetto di sviluppo on-premise in relazione all'evoluzione rappresentata dal cloud computing, mettendo in evidenza le caratteristiche, le sfide e i benefici di entrambi gli approcci. Nel panorama in costante evoluzione dello sviluppo software, il paradigma on-premise ha a lungo dominato la scena come l'approccio tradizionale per la creazione e la gestione di applicazioni all'interno dell'infrastruttura aziendale. Tuttavia, con l'emergere del cloud computing, si è aperta una nuova dimensione nel modo in cui le organizzazioni concepiscono e gestiscono le risorse informatiche.

Il cuore dell'approccio on-premise risiede nell'idea di sviluppare, implementare e gestire applicazioni software all'interno dell'ambiente hardware e software aziendale, piuttosto che fare affidamento su risorse esterne o soluzioni basate su cloud. In altre parole, le applicazioni sviluppate on-premise trovano sede fisica all'interno dell'organizzazione e sono gestite localmente dalle squadre IT aziendali. Un tratto distintivo è quindi il controllo locale che offre. Le aziende mantengono un controllo totale sull'intero ciclo di vita delle applicazioni, dalla fase di progettazione e sviluppo all'implementazione e alla manutenzione continua. Ciò significa che ogni aspetto del software è soggetto a un controllo diretto all'interno dell'organizzazione. L'aspetto della personalizzazione è, infatti, uno dei punti di forza dell'approccio on-premise. Le aziende possono adattare le applicazioni in base alle loro specifiche esigenze, incorporando funzionalità personalizzate e rispettando politiche di sicurezza altamente specifiche. Questa personalizzazione estrema è particolarmente preziosa quando le applicazioni devono soddisfare requisiti unici. Ad esempio, dal punto di vista della sicurezza e della conformità normativa, l'approccio on-premise offre un elevato grado di controllo. Poiché le applicazioni e i

dati risiedono all'interno del perimetro aziendale, le organizzazioni possono mantenere un controllo più rigoroso sulla sicurezza dei dati e garantire la conformità normativa. Questo è particolarmente rilevante in settori altamente regolamentati, come la sanità o le istituzioni finanziarie, dove la protezione dei dati e la conformità sono di primaria importanza. Tuttavia, va notato che lo sviluppo on-premise comporta investimenti iniziali sostanziali. Le aziende devono acquisire e gestire l'infrastruttura hardware e software necessaria per lo sviluppo e la manutenzione delle applicazioni in loco. Inoltre, la gestione delle applicazioni on-premise richiede la pianificazione e l'esecuzione di aggiornamenti e manutenzione personalizzati, incluso il monitoraggio delle patch di sicurezza e la gestione delle risorse hardware.

Dopo aver esaminato l'approccio on-premise, è cruciale comprendere come il cloud computing abbia rivoluzionato il modo in cui le organizzazioni concepiscono e gestiscono le risorse informatiche. Analizzeremo ora i vantaggi principali che il cloud computing offre rispetto al modello tradizionale on-premise, evidenziando come questi benefici abbiano reso il cloud una scelta preferita per molte aziende.
Una delle caratteristiche più evidenti del cloud computing è la sua notevole scalabilità. Le risorse informatiche, come la potenza di calcolo, lo spazio di archiviazione e la capacità di rete, possono essere facilmente e rapidamente scalate in base alle necessità aziendali in tempo reale. Questa flessibilità permette alle aziende di adattarsi agilmente a picchi di carico stagionali o a cambiamenti repentini nelle esigenze operative, senza dover effettuare costosi investimenti in hardware aggiuntivo.
Il modello di pagamento basato su consumo del cloud ha cambiato il panorama finanziario aziendale. Le organizzazioni possono evitare investimenti iniziali significativi in hardware e software, optando invece per un approccio in cui pagano solo per le risorse effettivamente utilizzate. Questo modello riduce i costi operativi e consente alle aziende di pianificare in modo più preciso i budget IT, contribuendo a una gestione dei costi più efficiente.
Il cloud computing offre un vantaggio decisivo in termini di accessibilità. Gli utenti possono accedere alle applicazioni, ai dati e ai servizi da qualsiasi luogo con una connessione Internet. Questa caratteristica è stata particolarmente rilevante nell'agevolare il lavoro remoto e la collaborazione tra gruppi sparsi in diverse sedi geografiche. È diventato più semplice per le aziende supportare modelli di lavoro flessibili e distribuiti.
Molte delle responsabilità di manutenzione hardware e software sono trasferite al provider di servizi cloud. Questo significa che le organizzazioni possono alleggerire il carico di lavoro del proprio personale IT, consentendo loro di concentrarsi su attività strategiche e progetti di valore aggiunto anziché dedicare tempo alla manutenzione operativa.
I servizi cloud spesso includono soluzioni di disaster recovery e business continuity avanzate. I dati vengono replicati su server geograficamente diversi, garantendo la disponibilità continua delle informazioni anche in caso di eventi catastrofici o

guasti hardware. Questo livello di resilienza è spesso difficile da ottenere con un approccio on-premise.
I provider di servizi cloud gestiscono gli aggiornamenti software in modo automatico, assicurandosi che le applicazioni siano costantemente aggiornate e sicure. Ciò elimina la necessità di pianificare e gestire manualmente gli aggiornamenti, riducendo il rischio di vulnerabilità legata a versioni obsolete.
Il cloud computing offre strumenti avanzati per la condivisione e la collaborazione tra team. Gli utenti possono lavorare contemporaneamente su documenti, progetti e dati in un ambiente condiviso, migliorando l'efficienza operativa e la produttività delle squadre.

In sintesi, il cloud computing rappresenta un cambiamento epocale rispetto all'approccio on-premise, grazie alla scalabilità, alla flessibilità dei costi, all'accessibilità da qualsiasi luogo, alla semplificazione della manutenzione, alla resilienza del disaster recovery, agli aggiornamenti automatici e al supporto alla collaborazione.

1.2 Microsoft Azure

Per esigenze legate cliente si è deciso di implementare l'applicazione con la piattaforma Microsoft Azure, ma gli stessi risultati si sarebbero potuti ottenere utilizzando diverse piattaforme come Amazon Web Services e Google Cloud Platform. Microsoft Azure rappresenta la piattaforma di cloud computing pubblico di Microsoft, offrendo una vasta gamma di servizi e vantaggi specifici, tra cui calcolo, analisi dati, archiviazione di alta qualità e networking efficiente. Gli utenti di Azure godono della massima libertà nel selezionare e utilizzare i servizi di cui hanno bisogno per soddisfare le loro esigenze. Possono scegliere tra questi servizi per sviluppare e scalare nuove applicazioni nel cloud pubblico o eseguire applicazioni esistenti. Con Azure, le aziende e le organizzazioni hanno la libertà di utilizzare gli strumenti e i framework che preferiscono per creare, gestire e distribuire applicazioni su una vasta rete globale.

Il sito web di Microsoft Azure offre un elenco di molti servizi diversi tra cui scegliere, tra cui macchine virtuali complete, database, archiviazione di file, backup e servizi per applicazioni web o mobile. È stato creato nel 2008 e ufficialmente rilasciato nel febbraio 2010. Originariamente noto come "Windows Azure," il nome è stato successivamente modificato in "Microsoft Azure" per riflettere la sua capacità di gestire non solo sistemi Windows, ma anche sistemi basati su Linux, consentendo agli utenti di eseguire macchine virtuali per entrambe le piattaforme.

Nel contesto di Microsoft Azure, si aprono due approcci fondamentali per sfruttare le risorse di cloud computing: Platform as a Service (PaaS) e Infrastructure as a Service (IaaS). Questi modelli di servizio offrono opzioni diverse per le aziende e

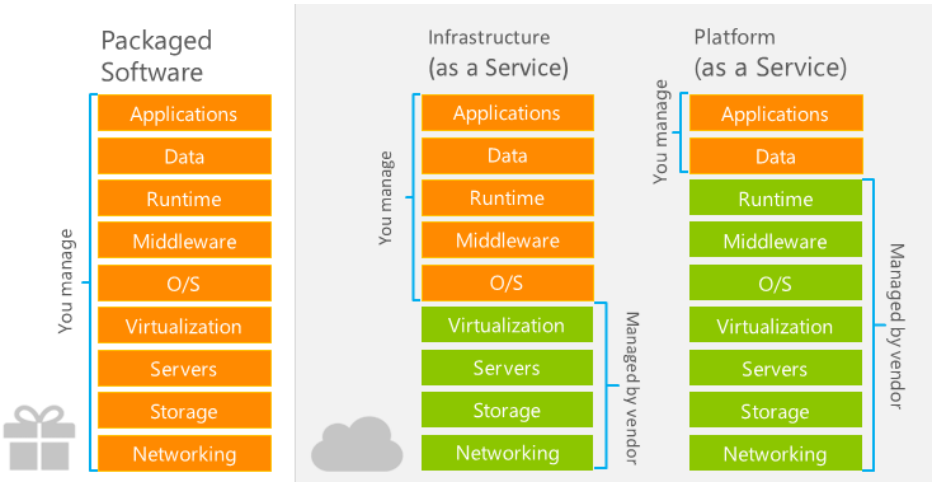

Figura 1.2: Confronto tra architettura on-premise, cloud Infrastructure as a Service (IaaS) e cloud Platform as a Service (PaaS).

gli sviluppatori, consentendo loro di selezionare l'approccio più adatto alle proprie esigenze e obiettivi.

- Platform as a Service, o PaaS, è un modello di servizio che offre un ambiente di sviluppo completo e gestito per la creazione, la distribuzione e la gestione di applicazioni. In questo contesto, Azure fornisce una piattaforma su cui gli sviluppatori possono costruire e ospitare applicazioni senza preoccuparsi della complessità sottostante dell'infrastruttura. Con PaaS, gli sviluppatori possono concentrarsi esclusivamente sullo sviluppo delle applicazioni, mentre l'infrastruttura, come il provisioning di server e risorse, viene gestita automaticamente da Azure. Azure PaaS consente di scalare rapidamente le risorse in base alle esigenze, sia in termini di capacità di calcolo che di archiviazione, per gestire picchi di carico o cambiamenti nella domanda. La gestione dell'ambiente operativo, inclusi aggiornamenti e patch, è responsabilità di Azure, riducendo il carico amministrativo per l'azienda.

- Infrastructure as a Service, o IaaS, fornisce risorse di calcolo, storage e networking in modalità on-demand all'interno di Azure. Con IaaS, gli utenti hanno il controllo completo su macchine virtuali e risorse di infrastruttura, ma possono ancora beneficiare dell'infrastruttura fisica gestita da Microsoft. Gli utenti hanno un maggiore controllo sulla configurazione delle macchine virtuali e delle risorse di rete, consentendo una maggiore personalizzazione delle soluzioni. IaaS offre la possibilità di eseguire applicazioni esistenti in Azure senza doverle riscrivere o adattare significativamente. In questo caso gli utenti sono responsabili della gestione e della manutenzione delle macchine

virtuali e delle risorse di infrastruttura, inclusi gli aggiornamenti del sistema operativo

La scelta tra PaaS e IaaS in Azure dipenderà dalle esigenze specifiche dell'azienda, dal livello di controllo desiderato e dalla natura delle applicazioni in sviluppo o in esecuzione. Entrambi i modelli offrono vantaggi distinti, consentendo alle aziende di adattare la loro strategia di cloud computing alle circostanze specifiche.

Data Factory

Azure Data Factory è un componente chiave all'interno dell'ecosistema di Microsoft Azure che svolge un ruolo fondamentale nella gestione e nell'elaborazione dei big data. Per comprendere appieno il suo ruolo, è importante iniziare con una definizione chiara. Azure Data Factory è un servizio completamente gestito e basato su cloud che consente di creare, pianificare e gestire flussi di lavoro di dati. Questi flussi di lavoro vengono utilizzati per l'acquisizione, la trasformazione e la distribuzione dei dati da e verso una vasta gamma di origini e destinazioni. In altre parole, Azure Data Factory agisce come un orchestratore dei dati, consentendo alle aziende di raccogliere, preparare e spostare dati in modo efficiente all'interno dell'ecosistema e oltre. Si tratta, infatti, di uno strumento chiave per gestire efficacemente sia la data ingestion che la data preparation notturna dei big data. Consentendo l'automazione, la scalabilità e il monitoraggio delle operazioni.

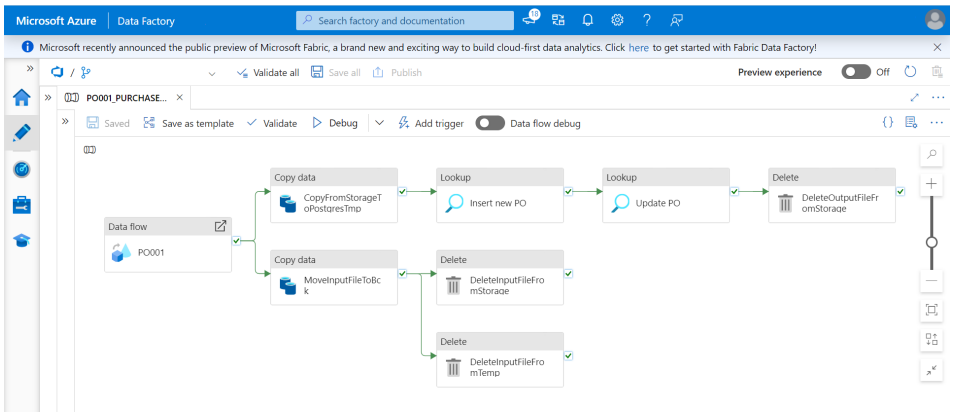

Figura 1.3: Esempio di una pipeline di Data Ingetion sviluppata in Azure Data Factory.

La data ingestion, nel contesto dei big data, rappresenta il processo di acquisizione dei dati da diverse fonti e la loro introduzione nel sistema di gestione dei dati. Questa fase iniziale è critica, in quanto i big data possono provenire da fonti eterogenee, tra cui dati strutturati e non strutturati, file di log, feed RSS, dati

dei sensori IoT e molto altro ancora. La gestione efficace della data ingestion è fondamentale per garantire che i dati siano disponibili per l'analisi e l'elaborazione.

La data preparation notturna dei big data è una pratica comune per ottimizzare l'utilizzo delle risorse e garantire che durante il giorno siano disponibili le risorse necessarie per le attività di business intelligence e analisi, riducendo al minimo il possibile impatto sulle operazioni aziendali. In questa fase notturna, alcune delle operazioni principali includono il filtraggio dei dati e l'aggregazione a vari livelli. Il filtraggio dei dati è una parte fondamentale della data preparation. Durante il giorno, i sistemi potrebbero acquisire dati grezzi provenienti da diverse fonti, che potrebbero contenere informazioni inutili o non pertinenti per le analisi future. La fase notturna può essere sfruttata per eseguire operazioni di filtraggio dei dati per eliminare le informazioni superflue e mantenere solo ciò che è rilevante per l'azienda. Azure Data Factory consente di automatizzare queste operazioni di filtraggio in base a criteri predefiniti, garantendo che solo i dati essenziali vengano conservati per l'analisi. L'aggregazione dei dati a vari livelli è un'altra operazione comune nella data preparation notturna. Questa operazione consente di riepilogare i dati a livelli più alti di granularità per velocizzare le future analisi. Ad esempio, è possibile aggregare dati giornalieri in dati mensili o annuali. Questo processo riduce il volume complessivo dei dati e accelera le query e l'analisi in seguito. Azure Data Factory consente di definire queste operazioni di aggregazione in modo che vengano eseguite automaticamente durante le ore notturne, quando le risorse sono più abbondanti e disponibili.

Inoltre, Azure Data Factory offre strumenti avanzati per il monitoraggio e il reporting dell'esecuzione dei flussi di lavoro di dati sia durante la data ingestion che durante la data preparation notturna. Gli amministratori possono verificare l'avanzamento delle operazioni, identificare eventuali problemi e generare report dettagliati sul processo di data preparation.

Kubernetes e Docker

La discussione su Kubernetes vs. Docker spesso si presenta come una scelta tra l'uno e l'altro, ma questa è una visione limitante. Un punto di forza di questo progetto è aver utilizzato insieme queste tecnologie diverse, che possono lavorare tramite scripting in modo complementare per costruire, distribuire e scalare applicazioni containerizzate.

Docker è una tecnologia open source e un formato di contenitore per automatizzare il rilascio di applicazioni come contenitori autosufficienti, portatili e utilizzabili sia nel cloud che in ambienti on-premise. Docker Inc. è una delle aziende principali che supporta la tecnologia Docker open source per eseguire applicazioni su Linux e Windows in collaborazione con provider cloud come Microsoft. Negli ultimi anni, Docker è diventato lo standard de facto per i container, offrendo un ambiente di

runtime per costruire e avviare container su qualsiasi macchina di sviluppo. Gli utenti possono quindi archiviare e condividere le immagini dei container tramite registri come Docker Hub o Azure Container Registry.

Kubernetes è un software open source di gestione che fornisce un'API per controllare come e dove i container verranno eseguiti. Kubernetes permette di eseguire i container Docker e gestire le operazioni complesse quando si tratta di scalare molti container distribuiti su server multipli. Questo strumento permette di orchestrare un cluster di macchine virtuali e pianificare l'esecuzione dei container su queste macchine in base alle risorse di calcolo disponibili e ai requisiti specifici di ciascun container. I container sono organizzati in POD, che rappresentano l'unità operativa di base di Kubernetes. Grazie a Kubernetes, è possibile scalare i container e i POD in base alle esigenze e gestire il loro ciclo di vita per garantire che le applicazioni rimangano operative. Mentre Docker offre un modo standard per confezionare e distribuire applicazioni containerizzate, Kubernetes arricchisce ulteriormente questa soluzione. Kubernetes offre funzionalità di orchestrazione avanzate, gestendo networking, bilanciamento del carico, sicurezza e scaling su tutti i nodi Kubernetes che eseguono i container. Dispone anche di meccanismi di isolamento come i "namespaces", che consentono di organizzare le risorse dei container in base all'accesso, all'ambiente di staging e altro ancora. In breve, Kubernetes e Docker lavorano in sinergia. Docker fornisce uno standard aperto per l'imballaggio e la distribuzione di applicazioni containerizzate, mentre Kubernetes offre l'orchestrazione avanzata necessaria per gestire complessi ambienti di container su larga scala. Questa collaborazione consente di rendere l'infrastruttura più resistente, l'applicazione più scalabile e di sfruttare appieno il potenziale delle applicazioni cloud-native.

Il processo di sviluppo e distribuzione del codice segue una pipeline (FIG 1.4) che inizia con lo sviluppo su DevOps e termina con l'implementazione su Azure Control System. Questa architettura modulare consente di sfruttare appieno le capacità del servizio di gestione dei container Kubernetes per distribuire il software in vari container, noti come POD. Questa suddivisione in container offre diversi vantaggi, inclusa una gestione modulare delle componenti dell'applicazione, semplificando gli aggiornamenti e la manutenzione. L'adozione di DevOps consente un ciclo di sviluppo continuo, agevolando l'implementazione di nuove funzionalità e correzioni di bug in modo rapido ed efficiente. Questa metodologia garantisce che l'applicazione rimanga sempre allineata alle esigenze aziendali e agli standard di prestazione.

Attraverso l'uso di container basati su Kubernetes, è possibile condurre simulazioni dettagliate che coinvolgono dati complessi e variabili, garantendo risultati affidabili e tempi di esecuzione efficienti. I container svolgono un ruolo importante nell'esecuzione di codice Python e Java dedicato alle simulazioni di performance e agli algoritmi di Machine Learning (ML) utilizzati per prevedere la scorta di pezzi

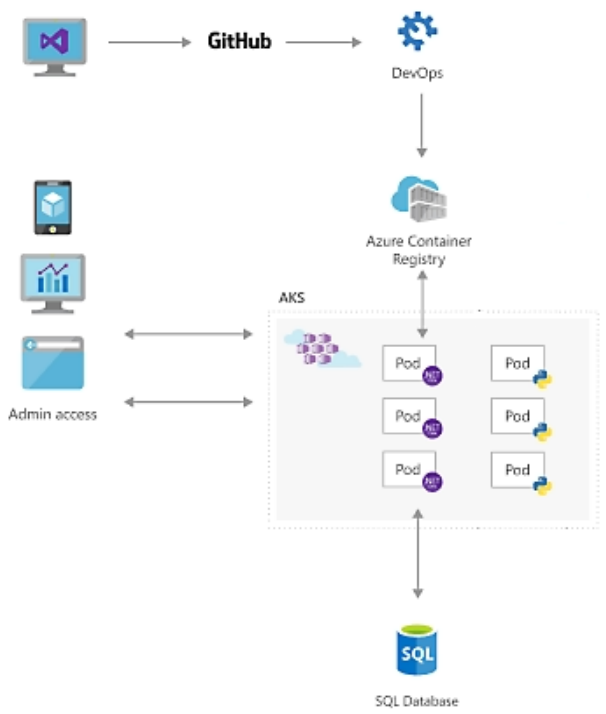

Figura 1.4: Il processo di sviluppo e distribuzione del codice segue una pipeline che inizia con lo sviluppo su DevOps e termina con l'implementazione su Azure Control System. Questa architettura permette di sfruttare le potenzialità del servizio di gestione dei container Kubernetes per distribuire il software in vari container, noti come POD.

di ricambio. Questa pratica è cruciale per ottimizzare la gestione dei magazzini di pezzi di ricambio e assicurare la continuità delle operazioni aziendali.

Per comprendere appieno il funzionamento di questo sistema, consideriamo un esempio concreto. Nell'immagine 1.5, possiamo osservare una pagina di dell'applicazione web sviluppata in questa tesi. Questa applicazione web è progettata per rispondere in modo interattivo con l'utente, e per farlo, utilizza una tecnologia denominata Angular, che esegue un codice specifico su un'entità chiamata POD. Nel contesto della creazione di una simulazione di scorta di sicurezza per i pezzi di ricambio in un magazzino, l'utente compie un'azione specifica, come ad esempio cliccare su un tasto di lancio (indicato dalla freccia rossa nell'immagine). Questa azione innescata dall'utente è il punto di partenza per l'esecuzione di una serie di processi complessi. In particolare, quando l'utente preme il tasto di lancio, un nuovo POD viene creato, si ricorda che il POD è un'unità di esecuzione nell'ambito di Kubernetes, il sistema di orchestrazione utilizzato. Questo nuovo POD conterrà

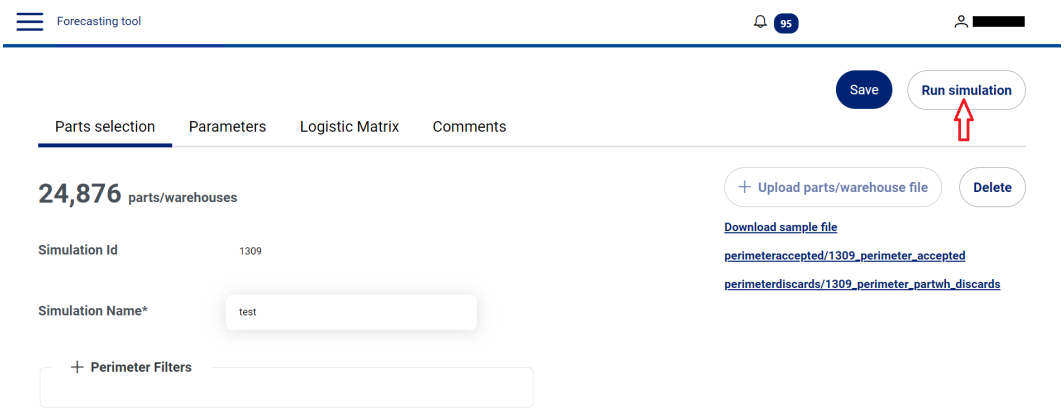

Figura 1.5: Un esempio pratico del funzionamento del sistema: una pagina di creazione di simulazioni di scorta di pezzi di ricambio in un magazzino. Cliccando sulla freccia rossa, viene creato un nuovo POD contenente il codice Python per eseguire calcoli complessi, con l'ausilio di Kubernetes per l'orchestrazione. Nel frattempo, il POD Angular rimane operativo per garantire una continua interattività con l'applicazione web.

il codice Python necessario per eseguire i calcoli richiesti per la simulazione. Il codice Python all'interno del POD interagisce con un database specifico, da cui legge tutti i dati di input preparati precedentemente. Questi dati sono stati raccolti e preparati durante la notte, come parte del processo notturno di data preparation. Il codice Python elabora questi dati secondo le specifiche richieste dalla simulazione, esegue i calcoli e scrive i risultati ottenuti nuovamente sul database così che possano poi essere visualizzati.

L'aspetto fondamentale è che Kubernetes orchestra l'intera operazione. Mentre il nuovo POD con il codice Python esegue i calcoli, il POD che ospita Angular rimane operativo, garantendo che l'applicazione web rimanga reattiva e utilizzabile per gli altri clienti. In altre parole, Kubernetes gestisce dinamicamente la creazione e la gestione dei POD in background, in modo che l'applicazione possa servire simultaneamente diversi utenti senza problemi di congestione o di performance. In questo modo, il sistema può eseguire elaborazioni complesse senza interruzioni nell'interazione dell'utente e garantendo una risposta veloce alle richieste

PowerBI

Nel mondo dell'analisi dei dati e della gestione delle performance aziendali, una delle sfide principali affrontate dalle organizzazioni è l'interpretazione dei dati. La

mole di informazioni disponibili può essere travolgente e complessa da comprendere senza l'uso adeguato di strumenti e tecniche di data visualization. In questo capitolo, esploreremo la difficoltà di interpretare i dati grezzi e l'importanza cruciale della data visualization per trasformare i dati in insight significativi. L'interpretazione errata dei dati può portare a decisioni aziendali sbagliate. Ad esempio, una lettura incompleta dei dati di performance dei magazzini potrebbe portare a una sovra-scorta o a una sottoscorta di pezzi di ricambio, con conseguenti impatti finanziari negativi. È fondamentale evitare errori costosi mediante una corretta interpretazione dei dati. Questi dati possono essere strutturati o non strutturati e possono variare notevolmente in formato e complessità. Interpretare questi dati grezzi richiede una comprensione approfondita delle metriche aziendali, ma anche una capacità di analisi sofisticata.

La data visualization è un potente strumento che ci consente di tradurre dati complessi in rappresentazioni visive, come grafici e dashboard interattive. È un linguaggio visuale che rende i dati aziendali più chiari e comprensibili rispetto alle fredde tabelle di dati grezzi. Questo approccio offre diversi vantaggi. Innanzitutto, le visualizzazioni possono rivelare tendenze, modelli e anomalie nei dati che altrimenti potrebbero passare inosservati. Quando i dati sono trasformati in forme visive, diventa più semplice individuare picchi, declini o comportamenti inusuali. Questa capacità di scoperta è fondamentale per prendere decisioni informate e guidare l'azione aziendale. Inoltre, le visualizzazioni sono un mezzo altamente efficace per comunicare informazioni complesse. Immaginate di dover presentare una vasta quantità di dati a un pubblico eterogeneo, composto da individui con diverse competenze e background. Le visualizzazioni semplificano notevolmente questo compito. Possono catturare l'attenzione, semplificare la comprensione e rendere più coinvolgente la condivisione di dati e risultati aziendali. Infine, una data visualization ben progettata contribuisce a prendere decisioni basate sui dati con maggiore fiducia. Quando le informazioni sono chiare e quando le tendenze emergono in modo evidente, le decisioni aziendali diventano più informate e basate su dati concreti.

Power BI è una potente piattaforma di business intelligence e data visualization sviluppata da Microsoft, si integra in modo nativo con Azure e altri servizi cloud, consentendo di accedere facilmente ai dati archiviati nei sistemi cloud dell'azienda. Questa integrazione semplifica l'analisi dei dati e l'elaborazione di informazioni in tempo reale. Power BI riveste un ruolo fondamentale nella comunicazione dei risultati aziendali attraverso dashboard interattive e strumenti avanzati per l'analisi dei Key Performance Indicators (KPI), tra cui il livello di servizio e il fill rate. Inoltre, permette l'automazione della generazione dei report, garantendo che i dati e le analisi siano sempre aggiornati. Questo riduce il carico di lavoro manuale e assicura che il personale aziendale disponga delle informazioni più recenti. Gli utenti possono monitorare in modo continuo questi indicatori chiave per valutare

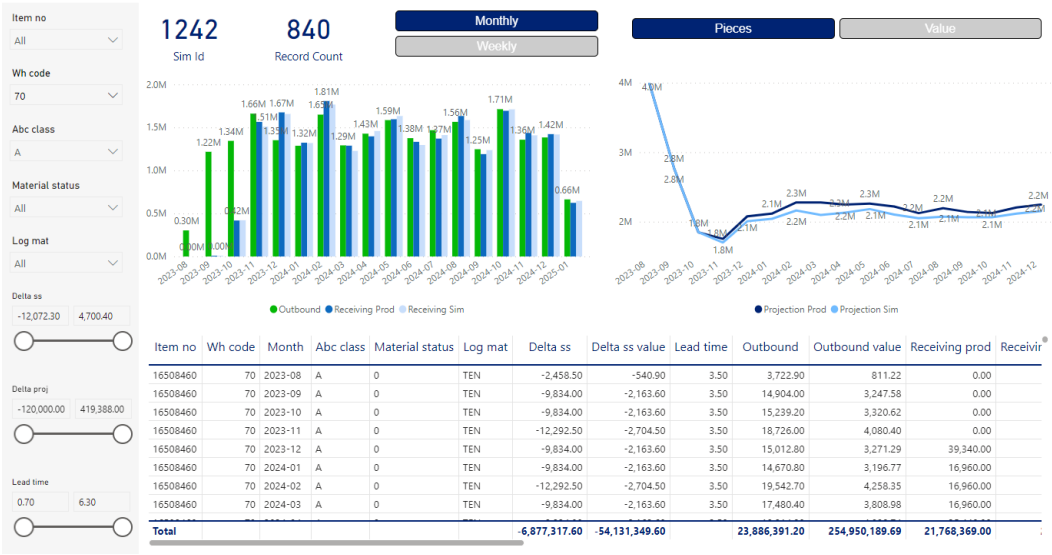

Figura 1.6: Dashboard di visualizzazione dei risultati di una simulazione.

l'efficacia delle operazioni dei magazzini e identificare eventuali aree di miglioramento. Le visualizzazioni dinamiche consentono di esaminare i dati da diverse prospettive e di identificare tendenze significative. Gli utenti possono selezionare le metriche e le visualizzazioni più rilevanti per il monitoraggio delle prestazioni dei magazzini di pezzi di ricambio. Questi dashboard possono essere condivisi con il personale chiave dell'azienda, consentendo una visualizzazione in tempo reale dei dati e delle analisi.

Nell'immagine 1.6 è mostrato un esempio di Dashboard realizzata tramite PowerBI sull'applicazione web. Qui, i dati vengono aggregati e rappresentati in vari formati, tra cui istogrammi, grafici a linee e tabelle. Sulla sinistra, è possibile filtrare i risultati in base a diversi parametri, permettendo un'analisi dettagliata e personalizzata. In basso, invece, è possibile visualizzare i dati nella tabella fino al massimo dettaglio, avendo come granularità il singolo pezzo di ricambio.

2 Modelli per Serie Temporali

Una serie temporale è una raccolta a intervalli di tempo regolari di osservazioni relative a diverse metriche. Argomento centrale di questa tesi è l'analisi delle serie temporali univariate che sono sequenza di misure relative a una singola variabile, in questo caso, la domanda mensile di uno specifico pezzo di ricambio. L'analisi delle serie temporali costituisce una gamma di strumenti e modelli che consentono di estrarre significato e conoscenza dai dati raccolti nel corso del tempo. Questi modelli sono in grado di svelare schemi, tendenze e comportamenti che potrebbero altrimenti sfuggire all'osservazione. Il loro utilizzo è essenziale per comprendere il flusso temporale dei dati e per ottenere informazioni preziose utili per prendere decisioni informate in una vasta gamma di settori.

Dataset

Il fulcro della nostra analisi di serie temporali è rappresentato da un insieme di dati, dove viene tracciato lo storico della domanda mensile di specifici componenti di ricambio, all'interno di un complesso sistema di gestione degli inventari. Il dataset è costituito da linee di domanda, che sono da intendersi come richieste specifiche per uno specifico pezzo di ricambio identificato da un codice univoco noto come "item_no". Inoltre, ogni richiesta è associata a un deposito specifico, noto come "warehouse". Questa associazione tra pezzi di ricambio e depositi è cruciale per comprendere il flusso di domanda all'interno di un sistema complesso di gestione degli inventari. È importante notare che nel contesto dell'applicazione web da cui proviene questo dataset, le combinazioni possibili di item_no e warehouse sono nell'ordine dei milioni, rappresentando una vasta gamma di scelte per gli utenti. Tuttavia, per i nostri scopi di analisi in questa tesi, è stata estratta una selezione più ristretta di circa 200.000 combinazioni item/warehouse. Questa selezione è stata basata su criteri che includono la disponibilità di uno storico di domanda significativo, permettendoci così di concentrarci su un sottogruppo di dati rappresentativo per l'analisi.

La serie temporale in questione è composta da osservazioni mensili della domanda di pezzi di ricambio. La lunghezza storica di questa serie copre un periodo di 5 anni, consentendo di disporre di dati per 4 anni come dataset di training. Il quinto anno è riservato come dataset di validazione delle logiche sviluppate, che svolge un ruolo cruciale nell'affinare e testare i modelli predittivi e le strategie di gestione dell'inventario. L'utilizzo di questo dataset di validazione è fondamentale

per garantire che le nostre analisi siano robuste e in grado di adattarsi alle variazioni stagionali e ai cambiamenti nei pattern di domanda che possono emergere nel corso del tempo. Inoltre, ci consente di valutare l'efficacia delle strategie di gestione dell'inventario sviluppate durante il corso della tesi.

Il punto di partenza dell'analisi di serie temporali è l'estrazione di caratteristiche significative dai dati al fine di creare un set di attributi informativi per la modellazione e l'analisi. Questo processo, noto come feature extraction, ci consente di condensare l'informazione contenuta nelle serie temporali in indicatori chiave che riflettono aspetti cruciali del comportamento della domanda, tra questi troviamo:

- Valore Medio della Domanda: Questo rappresenta la media delle osservazioni di domanda per un periodo di tempo specifico, fornendo una stima del livello medio di domanda.

- Varianza della Domanda: Misura quanto le osservazioni di domanda variano rispetto al loro valore medio, indicando la variabilità della domanda nel tempo.

- Coefficiente di Variazione (CV^2): è una misura relativa che esprime la varianza della domanda in rapporto al suo valore medio, fornendo una misura relativa della variabilità. Si calcola dividendo la varianza per il quadrato del valore medio.

- Intervallo tra le domande: Rappresenta l'intervallo di tempo tra due richieste consecutive del pezzo di ricambio, rivelando quanto tempo passa tra le occorrenze della domanda.

Nella figura 2.1, possiamo osservare una rappresentazione visuale dei dati presenti nel nostro dataset, focalizzandoci sugli attributi di intervallo tra le domande e CV^2. Questa rappresentazione mette in evidenza l'eccezionale varietà di comportamenti nel nostro dataset, riflettendo una vasta gamma di dinamiche di domanda per i pezzi di ricambio che stiamo esaminando. Un aspetto chiave che emerge dai dati è l'intervallo tra le domande. I valori rilevati evidenziano una notevole eterogeneità:

- Il valore minimo dell'intervallo tra le domande è 1, indicando che sono presenti dei pezzi per i quali c'è stata almeno un ordine per ogni mese. Questo sottolinea la presenza di pattern di domanda molto costanti, questi vengono definiti fast mover.

- D'altra parte, il valore massimo raggiunge 15.7, ciò significa che per questo pezzo di ricambio, in media, viene effettuato un ordine ogni 15 mesi. Questo

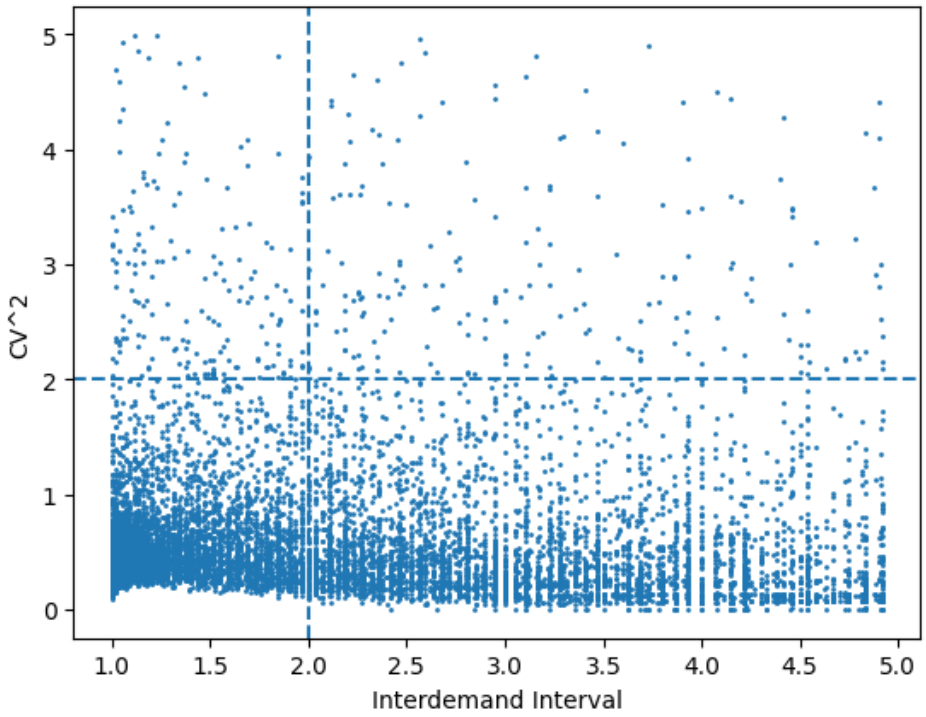

Figura 2.1: Distribuzione delle linee di domanda analizzate in termini di intervallo tra le domande e coefficiente di variazione.

intervallo considerevole riflette la presenza di pattern di domanda molto intermittenti, con significativi intervalli di tempo tra le richieste, questi pezzi vengono definiti slow mover.

La mediana di due mesi riveste il ruolo di valore critico. Sulla base di questo valore, classifichiamo le linee di domanda in fast mover e slow mover. Questa distinzione è fondamentale in quanto influenzerà notevolmente le strategie decisionali e le analisi future, poiché richiederà l'adozione di approcci specifici per la previsione della domanda e la gestione dell'inventario. D'altra parte, la distribuzione delle dimensioni della domanda è più uniforme, come indicato da una mediana del coefficiente di variazione CV^2 pari a 0.4. Notiamo inoltre che solo 10.000 linee di domanda superano il valore critico di 2.

L'estrazione di queste caratteristiche dal dataset ci fornisce un insieme di indicatori chiave che saranno fondamentali per le nostre analisi. Nel proseguo del capitolo, esploreremo come queste feature possono essere utilizzate per sviluppare modelli predittivi più avanzati.

2.1 Modelli analizzati

Di seguito, esploreremo nel dettaglio una serie di modelli, le loro varianti e le relative implementazioni, fornendo una panoramica completa delle metodologie di modellazione delle serie temporali. Tra i vari approcci, quattro spiccano per la loro ampiezza di applicazione e capacità di analisi.

1. MSTL (Multiple Seasonal and Trend decomposition using Loess) è una tecnica di decomposizione delle serie temporali che consente di separare e analizzare diverse componenti, come la tendenza e le stagionalità, attraverso l'utilizzo di smoothing Loess. Questo metodo è particolarmente utile quando una serie temporale presenta molteplici pattern stagionali di diversa frequenza, consentendo di individuare e comprendere tali variazioni in modo accurato.

2. ETS (Error, Trend, Seasonality) è una classe di modelli di previsione delle serie temporali che suddivide una serie in tre componenti principali: l'errore, la tendenza e la stagionalità. Questi modelli considerano l'errore come una componente casuale, la tendenza come una componente che rappresenta le variazioni a lungo termine e la stagionalità come le variazioni cicliche che si ripetono in determinati periodi. ETS è flessibile e può adattarsi a una varietà di pattern nei dati, rendendolo un'opzione versatile per l'analisi e la previsione delle serie temporali

3. I modelli ARIMA (Auto-Regressive Integrated Moving Average) sono strumenti fondamentali per l'analisi e la previsione delle serie temporali. Questi modelli collegano il valore attuale di una serie con i suoi valori passati attraverso un effetto auto-regressivo. L'integrazione rende stazionari i dati, mentre la media mobile considera gli errori di previsione passati per influenzare le future previsioni. ARIMA è ampiamente utilizzato per rivelare pattern e prevedere tendenze in una varietà di settori.

4. I modelli Croston rappresentano un approccio analitico progettato per affrontare le sfide delle serie temporali intermittentemente. In particolare, quando i dati presentano periodi in cui le osservazioni sono nulle o molto basse seguite da valori rilevanti. Questo tipo di serie temporali presenta una natura irregolare e può essere difficile da trattare con i metodi tradizionali. I modelli Croston mirano a catturare e prevedere queste irregolarità, sfruttando tecniche specifiche per gestire le fluttuazioni e fornire previsioni significative, anche in presenza di dati mancanti o irregolari.

Ciascun approccio offre strumenti preziosi per estrarre significato dalle serie temporali e prevedere andamenti futuri, sia attraverso relazioni causali, sia catturando schemi irregolari o stagionali.

2.1.1 Decomposizione in Trend e Stagionalità

Viene affrontato ora un aspetto cruciale dell'analisi delle serie temporali: la decomposizione in tendenza e stagionalità. Questa tecnica fondamentale ci consente di scomporre una serie temporale complessa in componenti più semplici e interpretabili, rivelando gli elementi sottostanti che guidano le variazioni nel tempo. Attraverso l'identificazione di queste componenti, siamo in grado di acquisire una comprensione approfondita delle dinamiche che influenzano i dati e delle tendenze che caratterizzano il lungo termine. Questo approccio arricchisce la nostra visione delle serie temporali e allo stesso tempo fornisce un solido fondamento per la creazione di previsioni più accurate e la formulazione di decisioni basate su dati concreti.

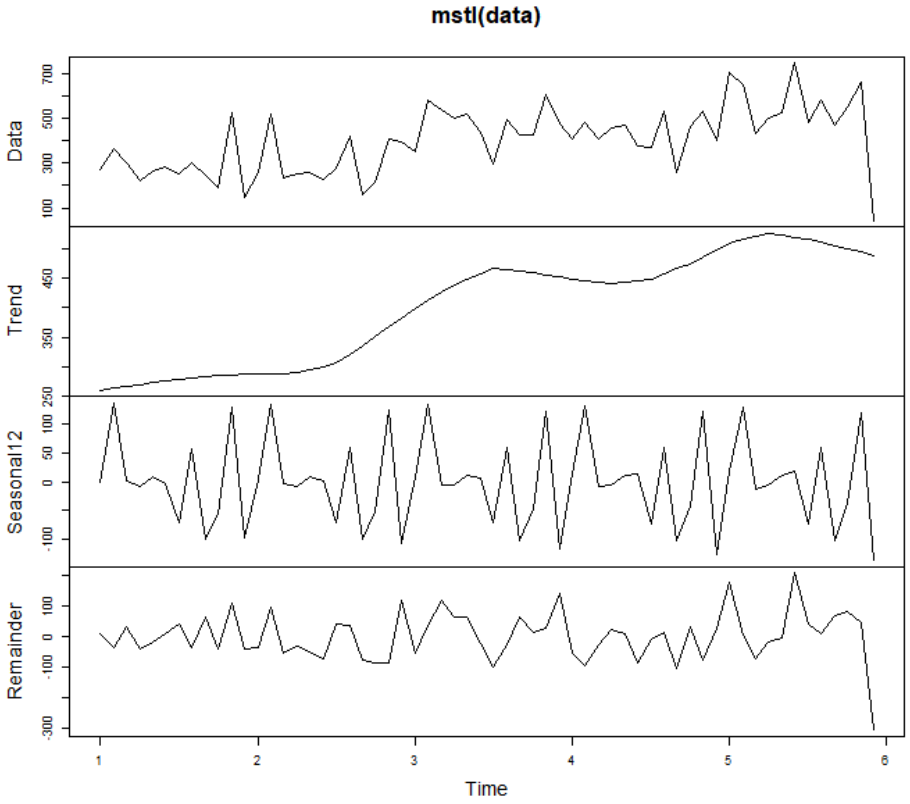

Figura 2.2: Un esempio di decomposizione in tendenza e stagionalità a 12 mesi.

Assumendo una decomposizione additiva, possiamo scrivere

$$y_t = S_t + T_t + R_t,$$

Dove:

- T_t, la componente di tendenza al tempo t, che riflette l'andamento di lungo termine della serie. Una tendenza è presente quando emerge una direzione costante di aumento o diminuzione nei dati. La componente di tendenza non deve essere necessariamente lineare.
- S_t, la componente stagionale al tempo t, che manifesta la stagionalità. Una forma stagionale è evidente quando la serie temporale è influenzata da fattori stagionali. Questi si manifestano in periodi fissi e noti (ad esempio, il trimestre dell'anno, il mese o il giorno della settimana).
- R_t, la componente irregolare (o "rumore") al tempo t, che rappresenta influenze casuali e disomogenee. Questo componente rappresenta il residuo o quanto rimane dalla serie temporale una volta rimosse le altre componenti.

Questo strumento si è rivelato cruciale durante l'analisi, consentendoci di comprender meglio il problema e sviluppare soluzioni alternative. Nel prosieguo di questo capitolo, esploreremo come l'utilizzo della decomposizione MSTL abbia contribuito significativamente a migliorare l'accuratezza delle previsioni che abbiamo formulato.

Filtraggio della domanda

Un'applicazione significativa della decomposizione è il filtraggio della domanda, una tecnica di pulizia dei dati finalizzata a migliorare l'efficacia delle previsioni. Il processo inizia con l'utilizzo della decomposizione MSTL (Multiple Seasonal Decomposition of Time Series), un metodo robusto che stima inizialmente le componenti di trend e stagionalità. MSTL viene applicato iterativamente ai dati al fine di ottenere stime accurate delle componenti stagionali. Questa fase di decomposizione è cruciale poiché fornisce una fondamentale comprensione delle tendenze nascoste nei dati. Successivamente, ci concentriamo sulla misurazione dell'intensità della stagionalità attraverso

$$F_s = 1 - \frac{Var(y_t - \hat{T}_t - \hat{S}_t)}{Var(y_t - \hat{T}_t)}.$$

Se $F_s > 0.5$, ovvero se la varianza spiegata dalla stagionalità è superiore al 50%, allora viene calcolata una serie destagionalizzata

$$y_t^* = y_t - \hat{S}_t$$

Qui viene utilizzata una soglia di forza stagionale poiché l'estimatore di $\hat{S}_t$ rischia di sovraadattarsi e diventare molto rumoroso se la stagionalità sottostante è troppo debole (o assente), con il potenziale rischio di mascherare eventuali valori anomali

assorbendoli nella componente stagionale. Questo passaggio riveste un ruolo chiave nel valutare quanto le variazioni cicliche influenzino i dati. Se $Fs \leq 0.5$, allora settiamo semplicemente $y_t^* = y_t$.

La componente di trend T_t è stimata nuovamente a partire da y^* usando il Super Smoother di Friedman, uno stimatore di regressione non parametrico basato su una regressione lineare locale con larghezze di banda adattive. L'ultimo step consiste nel cercare valori anomali nella serie dei residui

$$\hat{R}_t = y_t^* - \hat{T}_t.$$

Se $Q1$ denota il 25° percentile e $Q3$ denota il 75° percentile dei valori di resto, allora l'intervallo interquartile è definito come $IQR = Q3 - Q1$. Le osservazioni vengono etichettate come valori anomali se sono inferiori a $Q1 - 3 \cdot IQR$ o superiori a $Q3 + 3 \cdot IQR$.

È interessante notare che l'applicazione di questo tipo di filtraggio basato sull'intervallo interquartile (IQR) risulta particolarmente efficace nell'aumentare le prestazioni degli algoritmi di previsione, soprattutto per quei pezzi di ricambio per i quali si osserva un intervallo medio tra le domande vicino al valore critico. Queste pezzi, denominati anche medium mover, possono essere particolarmente influenzati nelle previsioni, e rimuovendo valori anomali è possibile ottenere previsioni più accurate e stabili. Questo approccio di filtraggio contribuisce quindi a ridurre l'effetto di outlier e rumore nei dati, migliorando così la qualità delle previsioni effettuate dagli algoritmi di previsione.

2.1.2 Smoothing esponenziale

I metodi di smoothing esponenziale sono in uso fin dagli anni '50. La classificazione di questi metodi è stata ripetutamente modificata nel corso degli anni portando alla definizione di quindici metodi distinti, come illustrato nella tabella seguente.

Tabella 2.1: I quindici metodi di smoothing esponenziale

Trend Component	Seasonal Component		
	N (Nessuno)	A (Additivo)	M (Moltiplicativo)
N (Nessuno)	(N,N)	(N,A)	(N,M)
A (Additivo)	(A,N)	(A,A)	(A,M)
A_d (Additivo smorzato)	(A_d,N)	(A_d,A)	(A_d,M)
M (Moltiplicativo)	(M,N)	(M,A)	(M,M)
M_d (Moltiplicativo smorzato)	(M_d,N)	(M_d,A)	(M_d,M)

Alcuni di questi metodi sono più conosciuti con altri nomi. Ad esempio, il caso (N, N) descrive il metodo di smoothing esponenziale semplice (SES), il caso (A, N)

descrive il metodo lineare di Holt, e il caso (A_d, N) descrive il metodo di tendenza smorzata. Il metodo di Holt-Winters additivo è rappresentato dalla combinazione (A, A) e il metodo di Holt-Winters moltiplicativo è rappresentato dalla combinazione (A, M). Le altre combinazioni corrispondono a metodi analoghi, ma meno comunemente utilizzati.

Seguendo l'approccio di Hyndman and Khandakar [2008], andiamo ad indicare la serie temporale osservata come $y_1, y_2, ..., y_n$. Una previsione di y_{t+h} basata su tutti i dati fino al tempo t è indicata come $\hat{y}_{t+h|h}$. Per illustrare il metodo, forniamo le previsioni puntuali e le equazioni di aggiornamento per il metodo (A,A), il metodo additivo di Holt-Winters:

$$\begin{aligned}
\text{Level:}\quad & l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\
\text{Growth:}\quad & b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \\
\text{Seasonal:}\quad & s_t = \alpha(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \\
\text{Forecast:}\quad & \hat{y}_{t+h|h} = l_t + b_t h + s_{t-m+h_m^+}
\end{aligned}$$

Dove m rappresenta la lunghezza della stagionalità (nel nostro caso 12 mesi), l_t rappresenta il livello della serie, b_t denota la, s_t rappresenta la componente stagionale, $\hat{y}_{t+h|h}$ è la previsione per h periodi in avanti, e $h_m^+ = [(h-1) \bmod m]+1$. Per utilizzare questo metodo, sono necessari valori per gli stati iniziali l_0, b_0 e $s_{1-m}, ..., s_0$, e per i parametri di smorzamento α, β e γ. Tutti questi saranno stimati dai dati osservati. La figura 2.3 fornisce formule ricorsive per il calcolo delle previsioni puntuali di h periodi in avanti per tutti i metodi di smoothing esponenziale.

Trend	Seasonal		
	N	A	M
N	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t / \ell_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t s_{t-m+h_m^+}$
A	$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t + hb_t$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t / (\ell_{t-1} - b_{t-1})) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t-m+h_m^+}$
A_d	$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t / (\ell_{t-1} - \phi b_{t-1})) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t-m+h_m^+}$
M	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t b_t^h$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^h + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t / (\ell_{t-1}b_{t-1})) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^h s_{t-m+h_m^+}$
M_d	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h}$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}^{\phi}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t / \ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t / (\ell_{t-1}b_{t-1}^{\phi})) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} s_{t-m+h_m^+}$

Figura 2.3: Le formule per i calcoli ricorsivi e le previsioni puntuali.

Ai parametri da stimare già citati si aggiunge ϕ, inoltre vale $\phi_h = \phi + \phi^2 + ... + \phi^h$.

Per ciascun metodo di smoothing esponenziale nella Tabella 2.3 è possibile descrivere due possibili modelli, uno corrispondente a un modello con errori additivi e l'altro a un modello con errori moltiplicativi. Se gli stessi valori dei parametri vengono utilizzati, questi due modelli forniscono previsioni puntuali equivalenti, sebbene con intervalli di previsione diversi. Pertanto, esistono 30 potenziali modelli descritti in questa classificazione. Siamo attenti a distinguere i metodi di smoothing esponenziale dai modelli di stato di base. Un metodo di smoothing esponenziale è un algoritmo per la produzione solo di previsioni puntuali. Il modello stocastico di stato sottostante fornisce le stesse previsioni puntuali, ma fornisce anche un quadro per il calcolo degli intervalli di previsione e altre proprietà.

Per distinguere i modelli con errori additivi e moltiplicativi, aggiungiamo una lettera aggiuntiva all'inizio della notazione del metodo. La terna (E, T, S) si riferisce alle tre componenti: errore, tendenza e stagionalità. Quindi il modello $ETS(A, A, N)$ ha errori additivi, tendenza additiva e nessuna stagionalità, la notazione $ETS(\cdot, \cdot, \cdot)$ aiuta a ricordare l'ordine in cui vengono specificate le componenti. Una volta specificato un modello, possiamo studiare la distribuzione di probabilità dei valori futuri della serie e trovare, ad esempio, la media condizionata di una futura osservazione dato la conoscenza del passato. Lo denotiamo come $\mu_{t+h|t} = \mathbb{E}(y_{t+h} \mid x_t)$, dove x_t contiene le componenti non osservate come l_t, b_t e s_t. Per molti modelli, queste medie condizionate saranno identiche alle previsioni puntuali date nella Tabella 2.3, in modo che $\mu_{t+h|t} = \hat{y}_{t+h|h}$. Tuttavia, per altri modelli (quelli con tendenza moltiplicativa o stagionalità moltiplicativa), la media condizionata e la previsione puntuale differiranno leggermente per $h \geq 2$.

Esempio 1. *Modello con errore additivo: $ETS(A, A_d, N)$*
Sia $\mu_t = \hat{y}_{t+1|h} = \hat{y}_t = l_{t-1} + \phi b_{t-1} + \epsilon_t$ la previsione di uno step avanti di y_t, assumendo che si conoscano i valori di tutti i parametri. Inoltre, $\epsilon_t = y_t - \mu_t$ rappresenta l'errore di previsione al tempo t. Dalle equazioni nella Tabella 2.3, otteniamo che

$$\begin{aligned} y_t &= l_{t-1} + \phi b_{t-1} + \epsilon_t \\ l_t &= l_{t_1} + \phi b_{t-1} + \alpha \epsilon_t \\ b_t &= \phi b_{t-1} + \beta^*(l_t - l_{t-1} - \phi b_{t-1}) = \phi b_{t-1} + \alpha\beta^* \epsilon_t = \phi b_{t-1} + \beta \epsilon_t \end{aligned}$$

Le tre equazioni sopra costituiscono rappresentazione in spazio di fase del metodo $ETS(A, A_d, N)$. Possiamo scriverlo in notazione standard specificando il vettore di stati come $\mathbf{x}_t = (l_t, b_t)'$, il sistema precedente diventa quindi:

$$\begin{aligned} y_t &= \begin{bmatrix} 1 & \phi \end{bmatrix} \mathbf{x}_{t-1} + \epsilon_t \\ \mathbf{x}_t &= \begin{bmatrix} 1 & \phi \\ 0 & \phi \end{bmatrix} \mathbf{x}_{t-1} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix} - \epsilon_t \end{aligned}$$

Il modello è completamente specificato una volta specificata la distribuzione del termine di errore ϵ_t. Solitamente assumiamo che siano indipendenti e identicamente distribuiti, seguendo una distribuzione normale con media 0 e varianza σ^2, che scriviamo come $\epsilon_t \sim NID(0, \sigma^2)$.

Ci sono rappresentazioni in spazio di fase simili dei modelli per tutte e 30 le varianti di smoothing esponenziale. Il modello generale coinvolge un vettore di stato $\mathbf{x}_t = (l_t, b_t, s_t, s_{t-1}, ..., s_{t-m+1})'$ e le equazioni di spazio di stato sono della forma:

$$y_t = \omega(\mathbf{x}_{t-1}) + r(\mathbf{x}_{t-1})\epsilon_t$$
$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + g(\mathbf{x}_{t-1})\epsilon_t$$

Dove $\{\epsilon_t\}$ è un processo di rumore bianco gaussiano con media zero e varianza σ^2, e $\mu_t = w(x_{t-1})$. Il modello con errori additivi ha $r(\mathbf{x}_{t-1}) = 1$, in modo che $y_t = \mu_t + \varepsilon_t$. Il modello con errori moltiplicativi ha $r(\mathbf{x}_{t-1}) = \mu_t$, in modo che $y_t = \mu_t(1 + \epsilon_t)$. Quindi, $\epsilon_t = (y_t - \mu_t)/\mu_t$ è l'errore relativo per il modello moltiplicativo. Tutti i metodi nella Tabella 2.3 possono essere scritti in questa forma. In generale, però, non tutti modelli sono lineari, ciò può essere sfidante, specialmente quando si tratta di stima e previsione. Tuttavia, una dei molti vantaggi di questi modelli è che, anche nel caso non lineare, possiamo comunque calcolare previsioni, la verosimiglianza e gli intervalli di previsione senza richiedere uno sforzo maggiore rispetto a quello necessario per il modello ad errori additivi. Infatti, una delle principali vantaggi nell'uso di questa forma è che consente di separare la modellazione delle dinamiche (transizione di stato) e il processo di osservazione, rendendo possibile gestire le non linearità in modo più efficiente.

Alcune delle combinazioni di tendenza, stagionalità ed errore possono occasionalmente causare difficoltà numeriche. I modelli di errore moltiplicativo sono utili quando i dati sono strettamente positivi, ma non sono numericamente stabili quando i dati contengono zeri o valori negativi. Quindi, poichè in questo progetto le serie temporali strettamente positive rappresentano una piccola minoranza dei casi, si è deciso di applicare solo i sei modelli completamente additivi. Per poter selezionare il migliore di questi modelli si possono usare misure di accuratezza delle previsioni, come l'errore quadratico medio (MSE), a condizione che gli errori siano calcolati dai dati in un set di validazione e non dagli stessi dati utilizzati per l'addestramento del modello. Tuttavia, si è preferito implementare un metodo penalizzato basato sulla corrispondenza interna. Uno di questi approcci utilizza una verosimiglianza penalizzata come il Criterio di Informazione di Akaike (AIC):

$$AIC = L^*(\hat{\theta}, \hat{\mathbf{x}}_0) + 2q$$

Dove q è il numero di parametri in θ più il numero di stati liberi in $\mathbf{x}_0$, mentre $\hat{\theta}$ e $\hat{\mathbf{x}}_0$ rappresentano le stime di θ e $\mathbf{x}_0$.

L'AIC fornisce anche un metodo per selezionare tra i modelli con errori additivi e moltiplicativi. Le previsioni puntuali dei due modelli sono identiche, quindi le misure standard di accuratezza delle previsioni, come l'MSE o l'errore percentuale medio assoluto (MAPE), non sono in grado di selezionare tra i tipi di errore. L'AIC è in grado di selezionare tra i tipi di errore perché si basa sulla verosimiglianza anziché sulle previsioni a un passo.

2.1.3 ARIMA

Con ARIMA (acronimo di AutoRegressive Integrated Moving Average) si intende una particolare tipologia di modelli atti ad indagare serie storiche che presentano caratteristiche particolari. I processi ARIMA sono un particolare sottoinsieme dei processi ARMA in cui alcune delle radici del polinomio sull'operatore ritardo che descrive la componente autoregressiva hanno radice unitaria (ossia uguale ad 1), mentre le altre radici sono tutte in modulo maggiori di 1.

In formule, dati i dati di serie temporali Z_t dove t è un indice intero e Z_t sono numeri reali, un modello ARIMA(p', q) è dato da:

$$Z_t - \alpha_1 Z_{t-1} - \alpha_2 Z_{t-2} - \ldots - \alpha_{p'} Z_{t-p'} = \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \ldots - \theta_q \epsilon_{t-q},$$

o equivalentemente da:

$$(1 - \alpha_1 B - \alpha_2 B^2 - \ldots - \alpha_{p'} B^{p'}) Z_t = (1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q) \epsilon_t,$$

dove B è l'operatore di *lag*, gli α_i sono i parametri della parte autoregressiva del modello, i θ_i sono i parametri della parte di media mobile e ϵ_t sono termini di errore. I termini di errore ϵ_t sono generalmente considerati variabili casuali, campionate da una normale con media zero e varianza σ^2.

Supponiamo ora che il polinomio $\left(1 - \sum_{i=1}^{p'} \alpha_i B^i\right)$ abbia una un fattore $(1 - L)$ di molteplicità d. Allora può essere riscritto come:

$$\left(1 - \sum_{i=1}^{p'} \alpha_i B^i\right) = \left(1 - \sum_{i=1}^{p'-d} \varphi_i B^i\right) (1 - B)^d.$$

Un processo ARIMA(p,d,q) esprime questa proprietà di fattorizzazione polinomiale con $p = p' - d$, ed è dato da:

$$\left(1 - \sum_{i=1}^{p} \varphi_i B^i\right) (1 - B)^d Z_t = \left(1 + \sum_{i=1}^{q} \theta_i B^i\right) \varepsilon_t,$$

e può quindi essere considerato come un caso particolare di un processo ARMA($p+d$,q) avente il polinomio autoregressivo con d radici unitarie. (Per questa ragione,

nessun processo che è accuratamente descritto da un modello ARIMA con $d > 0$ è stazionario nel senso ampio.)

Un ostacolo comune nell'utilizzo dei modelli ARIMA per le previsioni è la soggettività e la difficoltà nel processo di selezione dell'ordine. Nel paper Hyndman and Khandakar [2008] è stato sviluppato un metodo per individuare i parametri appropriati, basato sull'utilizzo del criterio di informazione AIC (Akaike Information Criterion):

$$AIC = -2log(L) + 2(p + q + P + Q)$$

Dove L rappresenta la massimizzata verosimiglianza del modello adattato ai dati differenziati $(1 - B^s)^D (1 - B)^d Z_t$. Tuttavia, la verosimiglianza del modello completo per Z_t non è effettivamente definita, il che impedisce di confrontare i valori dell'AIC per diversi livelli di differenziazione.

Tuttavia, cercare di selezione i parametri d e D minimizzando l'AIC, porta spesso al rischio di introdurre un eccesso di differenziazione, il quale può compromettere le previsioni future e aumentare l'ampiezza degli intervalli di confidenza. Di conseguenza, nell'ottica di migliorare le capacità predittive, è consigliabile cercare di applicare il minor numero possibile di operazioni di differenziazione. Per determinare i valori ottimali di d e D, il test di Dickey-Fuller (ADF) è preferibile. Questo test verifica l'ipotesi nulla di presenza di una radice unitaria in un campione di serie temporale, consentendo di stabilire se sia necessario applicare ulteriori differenziazioni. L'algoritmo aggiungerà gradualmente differenziazioni finché l'ipotesi nulla di presenza di una radice unitaria non verrà respinta. Una volta ottenuta una serie differenziata stabile, sarà possibile identificare il miglior modello ARMA utilizzando l'AIC come guida.

Effetti degli outliers su ARIMA

Nell'ambito dell'analisi delle serie temporali con ARIMA, un aspetto cruciale da considerare è l'influenza degli outliers o valori anomali sulla nostra comprensione dei dati. Gli outliers, ovvero punti dati che si discostano significativamente dalla tendenza generale della serie, possono distorcere in modo significativo la nostra percezione delle dinamiche sottostanti. Questo capitolo esplorerà gli effetti degli outliers nell'analisi delle serie temporali, concentrandosi in particolare sull'idea che la presenza di ordini non ripetitivi possa complicare ulteriormente la situazione.

L'algoritmo implementato su basa sul lavoro di Chen and Liu [1992]. Si inizia considerando una serie temporale $\{Z_y\}$ che segue un processo ARIMA generale, definito come:

$$Z_t = \frac{\theta(B)}{\varphi(B)\xi(B)}\varepsilon_t, \quad t = 1...n, \tag{2.1}$$

dove n rappresenta il numero di osservazioni per la serie; $\theta(B)$, $\varphi(B)$ e $\xi(B)$ sono polinomi di B; tutte le radici di $\theta(B)$ e $\varphi(B)$ si trovano al di fuori del cerchio unitario; e tutte le radici di $\xi(B)$ sono sul cerchio unitario. Per descrivere una serie temporale influenzata da un evento non ripetitivo, si considera il modello seguente:

$$Z^* = Z + \omega \frac{A(B)}{G(B)H(B)} I_t(t_1), \tag{2.2}$$

dove Z_t segue un processo ARIMA generale descritto in (2.1), $I_t(t_1) = 1$ se $t = t_1$, e $I_t(t_1) = 1$ altrimenti. Qui, $I_t(t_1)$ è una funzione indicatrice dell'occorrenza dell'impatto di un outlier, t_1 rappresenta la possibile posizione sconosciuta dell'outlier, e ω e $\frac{A(B)}{G(B)H(B)}$ rappresentano l'entità e il pattern dinamico dell'effetto dell'outlier.

Di seguito, affrontiamo il problema di stima quando sia la posizione che il pattern dinamico non sono noti a priori. L'approccio consiste nel classificare l'effetto dell'outlier in quattro tipi, imponendo una struttura speciale su $\frac{A(B)}{G(B)H(B)}$. I tipi includono un outlier innovativo (IO), un outlier additivo (AO), una variazione temporanea (TC) e uno di level shift (LS). Le loro definizioni sono le seguenti:

$$\text{IO}: \quad \frac{A(B)}{G(B)H(B)} = \frac{\theta(B)}{\varphi(B)\xi(B)}, \tag{2.3}$$

$$\text{AO}: \quad \frac{A(B)}{G(B)H(B)} = 1, \tag{2.4}$$

$$\text{TC}: \quad \frac{A(B)}{G(B)H(B)} = \frac{1}{(1-\delta B)} \tag{2.5}$$

$$\text{LS}: \quad \frac{A(L)}{G(L)H(L)} = \frac{1}{(1-L)}. \tag{2.6}$$

Gli outliers di tipo additivo (AO) si manifestano come picchi isolati, temporanei e non influenzano in modo significativo la tendenza generale dei dati. Possono essere causati da eventi eccezionali. Le variazioni temporanea (TC) rappresentano picchi transitori che appaiono e scompaiono rapidamente nella serie storica, il loro effetto nel tempo dipende dal parametro δ, gli autori del paper consigliano di settare un valore $\delta = 0.7$. Questi cambiamenti repentini sono di breve durata e possono essere attribuiti a fluttuazioni casuali. Gli outliers di tipo level shift (LS) causano un improvviso spostamento nel livello medio della serie storica, indicando un cambiamento sostanziale nella media dei dati. Questi eventi possono essere causati da condizioni persistenti o cambiamenti strutturali nei dati. Gli outliers di tipo innovativo (IO) hanno un impatto significativo sulla serie storica, determinando cambiamenti improvvisi e senza precedenti nel comportamento dei dati.

Questi eventi sono spesso associati a innovazioni, interruzioni o situazioni eccezionali. Per poter meglio visualizzare, nella figura 2.4 sono mostrate graficamente alcuni di questi errori.

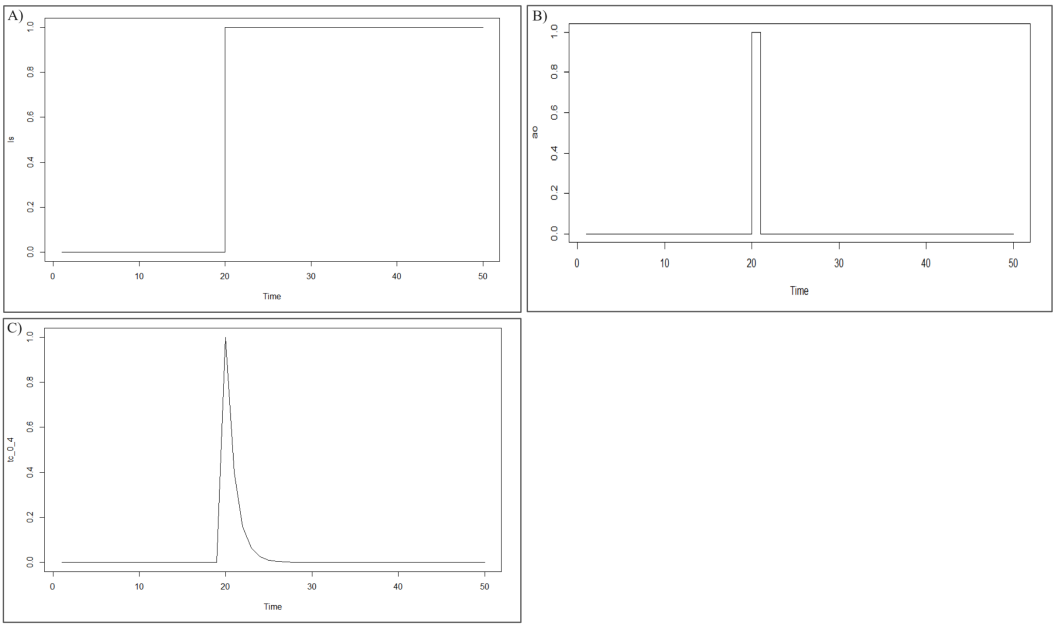

Figura 2.4: Esempi di vari tipi di outliers: A) spostamento di livello (LS), B) outlier additivo (AO), C) variazione temporanea (TC)

Nella pubblicazione Chen and Liu [1992], sono descritti in dettaglio tutti i passaggi di un algoritmo che consente di stimare simultaneamente i parametri di un modello ARIMA e i parametri associati all'errore. Per rendere più comprensibili i risultati ottenuti, si propone un esempio pratico dove la stima congiunta dei parametri è stata effettuata sfruttando il pacchetto "tsoutliers" disponibile su R.

Esempio 2. *Si consideri la serie temporale rappresentata nella figura 2.5.A. L'analisi condotta mediante l'algoritmo precedentemente descritto ha rivelato che questa serie segue un modello ARIMA*(0,1,2)(1,0,0)[12]. *In altre parole, la serie è stata sottoposta a una differenziazione di ordine uno, presenta una componente di media mobile di ordine 2 e una componente autoregressiva stagionale con un periodo stagionale di 12 mesi. Successivamente, è stato applicato l'algoritmo di stima congiunta dei parametri e degli errori, e i risultati ottenuti sono riportati nella Tabella 2.2. Inoltre, nella Figura 2.5.B è mostrata la serie filtrata (rappresentata dai pallini neri), evidenziando anche gli outliers rilevati (rappresentati dai pallini rossi). Inoltre, nella Tabella 2.2, emerge chiaramente un notevole miglioramento delle prestazioni del modello. Si può osservare che la varianza* σ^2 *subisce una significativa riduzione dopo il filtraggio della domanda. Questa diminuzione della*

variabilità indica una maggiore precisione e adattabilità del modello ai dati. Allo stesso tempo, i valori di verosimiglianza e AIC del modello mostrano un miglioramento significativo. Questo risultato suggerisce che il modello riesce a catturare in modo più accurato le relazioni nei dati, rendendolo una scelta migliore per l'analisi e la previsione dei fenomeni sottostanti. In generale, questi risultati evidenziano l'efficacia del processo di filtraggio nel miglioramento complessivo del modello.

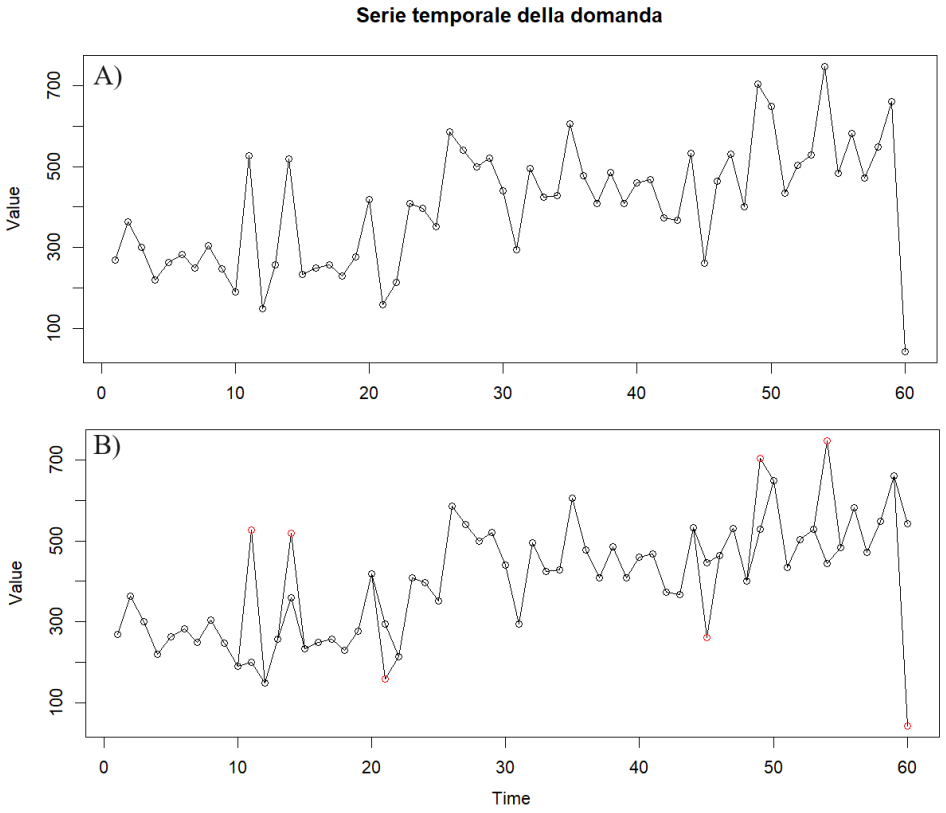

Figura 2.5: Serie temporale: i pallini neri rappresentano la serie temporale filtrata degli outliers rilevati dall'algoritmo, i pallini in rosso indicano i

Questo metodo si è dimostrato estremamente efficace nell'incrementare le performance dei modelli ARIMA. Dalle analisi condotte con il cliente è emerso che può portare a un notevole miglioramento nell'accuratezza delle previsioni, fino al 20%. Tuttavia, è importante notare che si tratta di un algoritmo iterativo che richiede un considerevole investimento di tempo per essere eseguito con successo. Di conseguenza, questo strumento risulta particolarmente utile per condurre analisi molto accurate su un insieme limitato di pezzi di ricambio particolarmente critici o rilevanti.

Tabella 2.2: Parametri del modello ARIMA(0,1,2)(1,0,0)[12]

Coefficienti	Stima Originale	Stima Congiunta
ma1	-0.8632	-0.0438
ma2	0.1430	-0.5465
sar1	0.6093	0.6093
AO11		326.6152
AO14		160.4151
AO21		-135.8759
AO45		-184.0687
AO49		175.0551
AO54		303.4378
AO60		-500.0719
$sigma^2$	16006	4962
Log Likelihood	-368.85	-332.38
AIC	745.7	686.76

Per l'analisi di big data sull'intero perimetro, è stata presa la decisione di implementare il modello ARIMA in una forma più semplice e meno intensiva dal punto di vista computazionale intensiva. Questa scelta consente di gestire un grande volume di dati in modo più efficiente, anche se a scapito di una precisione leggermente inferiore rispetto all'approccio iterativo. In definitiva, la strategia adottata permette di bilanciare la necessità di accuratezza con le limitazioni di risorse e tempo.

2.1.4 Croston

La domanda di tipo intermittente appare sporadicamente, con alcuni periodi di tempo in cui non vi è affatto domanda. Inoltre, quando si verifica la domanda, potrebbe non riguardare un'unità singola o una dimensione costante. Questa tipologia di domanda è difficile da prevedere e gli errori nelle previsioni potrebbero comportare costi in termini di magazzino o domanda insoddisfatta. Il Single Exponential Smoothing (SES) e la Simple Moving Averages (SMA) sono spesso utilizzati nella pratica per gestire la domanda intermittente. Entrambi i metodi hanno dimostrato di avere prestazioni soddisfacenti su dati reali di domanda intermittente. Tuttavia, il metodo di previsione standard per gli articoli con domanda intermittente è considerato il metodo di Croston. Croston ha costruito stime della domanda dagli elementi costituenti, ovvero la dimensione della domanda quando si verifica (z_t) e l'intervallo tra le domande (p_t).

Si assume che sia la dimensione della domanda che gli intervalli siano stazionari. Si suppone che la domanda si verifichi come un processo di Bernoulli. Di

conseguenza, gli intervalli tra le domande sono distribuiti geometricamente (con una media ρ). Si ipotizza che le dimensioni della domanda seguano la distribuzione normale (con una media μ e varianza σ^2). Secondo il metodo di Croston, attraverso uno smoothing esponenziale si effettuano stime separate per la dimensione media della domanda ($\tilde{z}_t, \mathbb{E}[z_t] = \mathbb{E}[\tilde{z}_t] = \mu$) e l'intervallo medio tra gli episodi di domanda ($\tilde{p}_t, \mathbb{E}[p_t] = \mathbb{E}[\tilde{p}_t] = \rho$) dopo che si è verificata la domanda (utilizzando lo stesso valore di costante di smoothing). Se non si verifica alcuna domanda, le stime rimangono esattamente le stesse. La previsione, $\tilde{Y}_t$, per il prossimo periodo di tempo è data da $\tilde{Y}_t = \tilde{z}_t/\tilde{p}_t$ e, secondo Croston, l'estimatore previsto della domanda per periodo in quel caso sarebbe $\mathbb{E}[\tilde{Y}_t] = \mathbb{E}[\tilde{z}_t/\tilde{p}_t] = \mathbb{E}[\tilde{z}_t]/\mathbb{E}[\tilde{p}_t] = \mu/\rho$ (cioè, il metodo è unbias). Si osserva che se la domanda si verifica in ogni periodo di tempo, l'estimatore di Croston è identico al SES. Il metodo è, almeno intuitivamente, superiore a SES e SMA.

È stato dismostrato che l'estimatore di Croston è soggetto a un bias. Il bias sorge perché, se si assume che gli estimatori della dimensione della domanda e dell'intervallo di domanda sono indipendenti, allora

$$\mathbb{E}\left[\frac{\tilde{z}_t}{\tilde{p}_t}\right] = \mathbb{E}[\tilde{z}_t] \cdot \mathbb{E}\left[\frac{1}{\tilde{p}_t}\right]$$

ma

$$\mathbb{E}\left[\frac{1}{\tilde{p}_t}\right] \neq \frac{1}{\mathbb{E}[\tilde{p}_t]}$$

e quindi il metodo di Croston è distorto. È evidente che questo risultato non dipende dalle assunzioni di stazionarietà e distribuzione geometrica degli intervalli di domanda di Croston. La dimensione dell'errore dipende dal valore della costante di smoothing utilizzato. È stato dimostarto che il bias associato al metodo di Croston può essere approssimato, per tutti i valori della costante di smoothing, da $\frac{\alpha}{2-\alpha}\mu\frac{\rho-1}{\rho^2}$ (dove α è il valore della costante di smoothing utilizzata per l'aggiornamento degli intervalli tra le domande).

Nel lavoro di Syntetos and Boylan [2005] viene proposto un nuovo metodo di previsione per la domanda intermittente, che incorpora questa approssimazione del bias. Il metodo è stato sviluppato basandosi sull'idea di Croston di costruire stime della domanda da elementi costituenti. Esso è approssimativamente privo di distorsioni ed è stato dimostrato che supera il metodo di Croston su dati generati teoricamente. Il nuovo stimatore della domanda media è il seguente:

$$\tilde{Y}_t = \left(1 - \frac{\alpha}{2}\right)\frac{\tilde{z}_t}{\tilde{p}_t}$$

dove α rappresenta il valore della costante di smoothing utilizzata per l'aggiornamento degli intervalli tra le richieste.

Tabella 2.3: DataFrame di Input dell'algoritmo di previsione della domanda.

item_no	*wh_code*	*n*	*repet_qty_01*	*repet_qty_02*
41272826	2	57	1.0	2.0
41284187	5	60	0.0	0.0
41284187	50	60	1.0	1.0
5949894564	16	53	4.0	0.0
8143852	15	60	2.0	3.0
5801288792	15	55	0.0	0.0

```
from statsforecast.models import CrostonSBA, CrostonClassic

models = [CrostonSBA(), CrostonClassic()]
aliases = ['CrostonSBA', 'CrostonClassic']
results = []
for id in ids:
    train = data.loc[data['unique_id'] == id,
       qty[int(data.loc[data['unique_id'] == id,
       'n'])-1:10:-1]].values[0]
    validation = data.loc[data['unique_id'] == id,
       qty[11::-1]].values[0]
    for model, alias in zip(models, aliases):
        model.fit(train)
        pred = model.predict(h=12)['mean']
        MSE = ((pred - validation) ** 2).mean()
        MAE = (abs(pred - validation)).mean()
        s = (abs(train[1:] - train[:-1])).mean()
        MASE = MAE / s
        result = {'unique_id': id, 'alias': alias, 'MAE': MAE,
           'MSE': MSE, 'MASE': MASE, 'prediction': pred,
           'values': validation}
        results.append(result)
```

Codice 2.1: Algoritmo che effettua training e validation degli algortimi Croston e CrostonSBA.

Esempio 3. *Abbiamo condotto una valutazione dell'applicabilità dei risultati teorici al nostro dataset di domande. In particolare, abbiamo selezionato un campione rappresentativo di domande, concentrando l'attenzione sui cosiddetti pezzi slow mover, ovvero quelli con un intervallo medio tra richieste superiore a due mesi. Nella Tabella 2.3, è possibile osservare la struttura del nostro dataset di input, in particolare item_no e wh_code identificano il codice del singolo pezzo e del magazzino che lo ha richiesto. La variabile n indica la lunghezza effettiva dello storico delle domande a partire dalla sua messa in produzione, mentre*

repet_qty_01, ..., repet_qty_60 rappresenta la serie storica delle domande numerate in ordine cronologico inverso. Nel Codice 2.1 si può leggere lo script relativo all'applicazione del metodo di Croston (denominato CrostonClassic) e del metodo proposto chiamato CrostonSBA. Abbiamo inoltre valutato diverse metriche, riscontrando che i risultati teorici si adattano bene al nostro specifico contesto di analisi. Nell'immagine 2.6, è visualizzata la distribuzione dei dati che sono stati predetti in modo più accurato da ciascuno dei due algoritmi in relazione all'intervallo tra le richieste. È evidente che, per ogni valore di Intervallo tra le domande, la percentuale di pezzi in cui CrostonSBA ottiene prestazioni superiori è notevolmente più alta. Questo suggerisce che il modello teorico è robusto per l'analisi delle nostre domande. Di conseguenza, abbiamo preso la decisione di applicare l'algoritmo di CrostonSBA a tutti i pezzi di ricambio classificati come slow mover, sia per sfruttarne le prestazioni migliori che per ottimizzare l'efficienza del nostro codice.

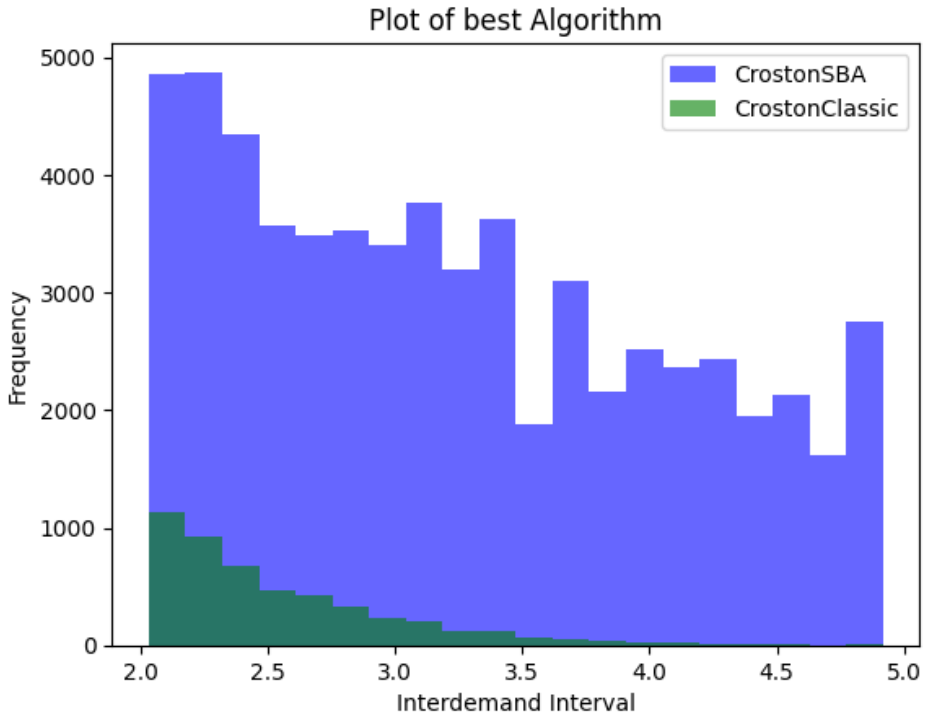

Figura 2.6: CrostonSBA vs. CrostonClassic

2.2 Implementazione degli algoritmi

Nel seguente capitolo, esploreremo in dettaglio l'implementazione degli algoritmi di previsione all'interno dell'applicazione web dedicata all'analisi e al miglioramento delle performance dei magazzini di pezzi di ricambio. Prima di addentrarci nei dettagli implementativi, è fondamentale acquisire una comprensione approfondita delle principali metriche utilizzate per valutare l'accuratezza delle previsioni. Ecco un'analisi dettagliata di queste metriche chiave:

- Il Mean Absolute Error (MAE) rappresenta una delle metriche fondamentali per valutare l'accuratezza delle previsioni. Questa misura calcola la media dei valori assoluti degli errori tra le previsioni e i valori osservati, ovvero i dati reali. La formula per il calcolo del MAE è la seguente:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

  dove:

  - n è il numero totale di osservazioni.
  - Y_i rappresenta il valore osservato per l'osservazione i.
  - $\hat{Y}_i$ rappresenta la previsione per l'osservazione i.

- Il Mean Absolute Scaled Error (MASE) è una metrica che mira a fornire una misura di errore relativa e scalabile. Questo lo rende particolarmente utile quando si desidera confrontare l'accuratezza delle previsioni tra diverse serie temporali o quando si cerca una metrica di errore robusta che non dipenda dalla scala dei dati. La formula per il calcolo del MASE è la seguente:

$$MASE = \frac{MAE}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|}$$

  Le variabili coinvolte sono le stesse del MAE, con l'aggiunta di Y_{i-1}, che rappresenta il valore osservato per l'osservazione precedente a Y_i

- Il Mean Squared Error (MSE) è un'altra metrica comune utilizzata per valutare l'accuratezza delle previsioni. Questa metrica calcola la media dei quadrati degli errori tra le previsioni e i valori osservati. La formula per il calcolo del MSE è la seguente:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Queste tre metriche svolgono un ruolo cruciale nell'analisi delle performance degli algoritmi di previsione implementati nell'applicazione web. Ciascuna metrica offre una prospettiva unica sull'accuratezza delle previsioni, consentendo di valutare quale algoritmo di previsione funzioni meglio in diverse situazioni di domanda all'interno dei magazzini di pezzi di ricambio.

Allenamento degli Algoritmi

Il primo passaggio di questo processo coinvolge il calcolo dell'intervallo medio tra le domande. Questo calcolo permette di raggruppare le domande in base al loro comportamento. Questa suddivisione è cruciale, poiché gli algoritmi implementati e le metriche di valutazione saranno specifici per ciascun cluster di domande. In particolare, si distinguono due categorie principali: i fast mover, caratterizzati da intervalli tra le domande più brevi, e gli slow mover, con intervalli più lunghi tra le domande. Dopo aver suddiviso l'insieme di dati storici in due parti, il processo di allenamento degli algoritmi di previsione entra nella sua fase chiave. Questa fase è essenziale per costruire modelli statistici o matematici in grado di fare previsioni accurate basate sui dati storici noti, aprendo la strada a proiezioni future.

Per quanto riguarda i fast mover, vengono utilizzati i modelli ETS (Error, Trend, Seasonality) e ARIMA (AutoRegressive Integrated Moving Average). La selezione di questi modelli avviene attraverso l'impiego delle funzioni 'autoETS' e 'AutoArima' presenti nella libreria statforecast. Queste funzioni automatizzate effettuano l'addestramento e la valutazione di una serie di varianti dei modelli ETS e ARIMA, ciascuna con diverse configurazioni. La scelta dei migliori modelli all'interno di ciascuna famiglia avviene sulla base del criterio AIC (Akaike Information Criterion), come dettagliato rispettivamente nelle sezioni 2.1.2 e 2.1.3. L'obiettivo finale è identificare il modello che si adatta meglio ai dati storici per ciascuna delle due famigle di algoritmi di forecast, e utilizzarlo per effettuare previsioni per un numero di mesi futuri pari alla lunghezza del dataset di validazione.

Per gli slow mover, invece, vengono addestrati gli algoritmi CrostonSBA, come illustrato nel capitolo corrispondente 2.1.4, e l'algoritmo di Simple Exponential Smoothing. Questi modelli sono adatti per gestire domande caratterizzate da una domanda intermittente o con un comportamento meno volatile rispetto ai "fast mover". L'addestramento di questi modelli prevede la stima dei parametri appropriati e la generazione di previsioni basate su tali parametri.

Validazione degli Algoritmi

Dopo aver completato i passaggi precedenti, otterremo due modelli di previsione distinti, ciascuno selezionato in base alle specifiche caratteristiche del cluster di

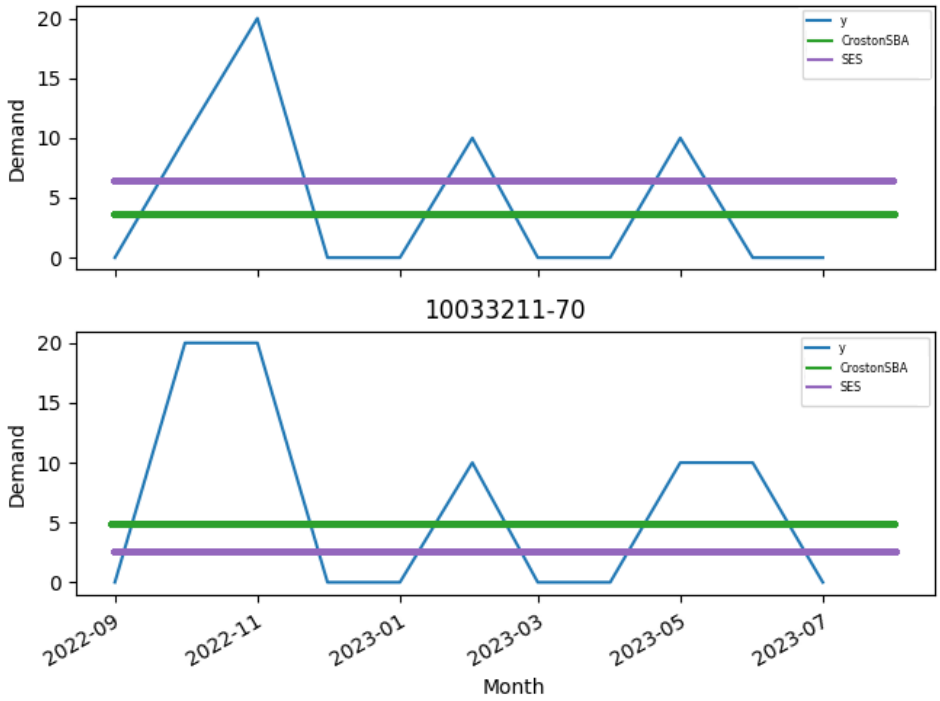

Figura 2.7: Confronto tra CrostonSBA e Simple Exponential Smoothing per le previsioni nel cluster Slow Mover: Una panoramica della validazione della precisione delle previsioni.

domanda corrispondente. Questi modelli saranno poi impiegati per generare previsioni future per ogni tipo di prodotto. La fase di validazione riveste un ruolo cruciale in questo processo poiché ciò che vogliamo è garantire che la scelta dell'algoritmo più efficace sia basata sulla sua capacità di fare previsioni accurate su un dataset indipendente da quello utilizzato per il training. Inoltre, nel processo di validazione si tiene conto delle specifiche esigenze di previsione legate al cluster della domanda, ciò consente di definire delle metriche specifiche per selezionare gli algoritmi di previsione più adatti.

Per quanto riguarda i pezzi clusterizzati come "fast mover," essi rappresentano prodotti ad alta rotazione con intervalli tra le domande relativamente brevi. In questo contesto, valutiamo l'efficacia degli algoritmi di forecast basandoci principalmente sull'errore quadratico medio (MSE). Questo approccio è progettato per privilegiare gli algoritmi che sono in grado di catturare con maggiore precisione le variazioni stagionali e le fluttuazioni di domanda di questi prodotti altamente dinamici.

D'altra parte, i pezzi classificati come "slow mover" rappresentano prodotti a basso rotazione, caratterizzati da intervalli tra le domande molto più lunghi. In

questo caso, la nostra preferenza va verso la scelta dell'algoritmo migliore basata sulla metrica del Mean Absolute Scaled Error (MASE). Questa metrica è particolarmente adatta a questo tipo di prodotti poiché favorisce gli algoritmi in grado di fornire previsioni che compensano in modo efficace e accurato le fluttuazioni di domanda su periodi temporali più ampi. L'obiettivo principale è garantire una previsione stabile e affidabile nel lungo termine, anche se la frequenza delle domande è bassa.

Previsione della Domanda Futura

Infine, dopo aver identificato il miglior algoritmo di forecast per ciascun pezzo analizzato, si procede con l'ultimo step fondamentale. In questa fase, l'intero dataset, che comprende sia i dati di training che quelli di validazione, viene utilizzato per addestrare il modello selezionato. Una volta completato questo passaggio, si dispone di un modello che è stato calibrato per riflettere al meglio le caratteristiche della domanda passata.

A questo punto, il modello può essere utilizzato per generare previsioni della domanda per tutti i mesi futuri. Queste previsioni sono fondamentali per la pianificazione e la gestione della catena di approvvigionamento, consentendo di ottimizzare gli stock e le risorse in modo più efficiente. Inoltre, un output altrettanto significativo di questo processo è la stima della deviazione standard della domanda futura. Questo valore fornisce un'indicazione preziosa sulla variabilità prevista nella domanda, aiutando a definire livelli di scorta appropriati e a mitigare il rischio di eccessi o mancanze di inventario.

Tutte queste informazioni saranno fondamentali nel prossimo capitolo, in cui verranno calcolati e analizzati indicatori essenziali per la gestione del magazzino. Sarà un momento chiave per tradurre le previsioni in decisioni concrete e pianificazioni operative per garantire un flusso efficiente nella catena di approvvigionamento.

3 Modelli di Gestione degli Inventari

Il ruolo del magazzino all'interno di un'azienda è di fondamentale importanza nel contesto della gestione delle scorte e delle risorse logistiche. Qui convergono materie prime, prodotti finiti e componenti chiave, rappresentando un punto cruciale per il successo operativo e finanziario dell'azienda. Un'efficiente gestione del magazzino non solo garantisce la disponibilità costante dei prodotti, ma può anche contribuire a ridurre i costi, migliorare i tempi di consegna e, di conseguenza, soddisfare in modo più completo le aspettative dei clienti. Nel capitolo seguente, esploreremo dettagliatamente diverse strategie di gestione del magazzino, fornendo un'ampia panoramica delle migliori pratiche, delle tecniche avanzate e delle soluzioni tecnologiche che possono essere utilizzate per ottimizzare l'efficienza del magazzino e massimizzare la redditività aziendale. Dalle metodologie tradizionali, come il "just-in-time" e il "modello economico dell'ordine", alle soluzioni più moderne, come i sistemi di gestione dell'inventario computerizzati e l'automazione dei processi, esamineremo come ciascuna di queste strategie possa essere adattata alle specifiche esigenze di diverse industrie e settori.

3.1 Analisi e Scelta della Strategia di Gestione dell'Inventario nel Settore Automotive

Approccio Just-in-Time

Il concetto cardine dell'approccio Just-in-Time (JIT) è la minimizzazione dell'inventario, mantenendo solamente ciò di cui si ha effettivamente bisogno al momento opportuno. In un'ottica ideale, si mira a gestire un magazzino che riceve esattamente la quantità di prodotti previsti per essere venduti nel breve termine. Questo approccio mira a ridurre al minimo il capitale immobilizzato negli inventari e i costi associati allo stoccaggio a lungo termine. Per implementare il JIT, è necessaria una pianificazione della produzione precisa e una stretta collaborazione con i fornitori. Gli ordini vengono effettuati solo quando le scorte stanno per esaurirsi o quando c'è una domanda effettiva. Questo metodo richiede una gestione eccellente della catena di approvvigionamento per garantire la disponibilità delle parti o dei prodotti giusti nel momento giusto. L'obiettivo è evitare eccessi di inventario e sprechi, contribuendo così a migliorare l'efficienza operativa.

Modello del Lotto Economico

Il Modello del Lotto Economico (EOQ) rappresenta una strategia basata su calcoli matematici volta a determinare la quantità ottimale da ordinare con l'obiettivo di minimizzare i costi totali. Tali costi comprendono sia i costi fissi di ordinazione, come le spese per la gestione degli ordini, che i costi variabili associati al mantenimento delle scorte, come i costi di stoccaggio. L'obiettivo principale consiste nel trovare un punto di equilibrio tra questi due tipi di costi, al fine di ridurre al minimo l'importo complessivo impiegato nella gestione dell'inventario. L'EOQ fa uso di una formula specifica per calcolare la quantità ottimale da ordinare, considerando la domanda annuale, i costi di ordine e i costi di mantenimento delle scorte. Una volta ottenuto questo valore ottimale, l'azienda può stabilire il momento e la quantità degli ordini in modo da massimizzare l'efficienza e allo stesso tempo ridurre i costi. Questo approccio, sebbene possa apparire complesso inizialmente, fornisce una struttura chiara per prendere decisioni in merito alle quantità di ordini all'interno del contesto della gestione degli approvvigionamenti. Per illustrare il concetto di EOQ e la sua formula principale, fornirò un esempio semplificato.

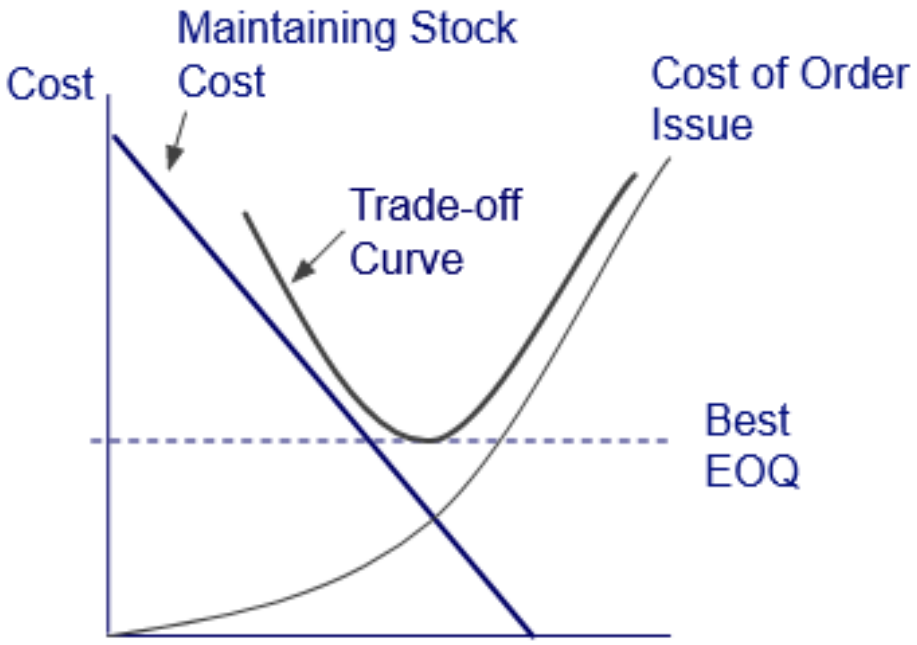

Figura 3.1: Calcolo EOQ: Trovando l'equilibrio tra costi di ordinazione e costi di stoccaggio.

Esempio 4. *Il modello base del lotto economico si fonda su alcune ipotesi fondamentali:*

- *Prodotti indipendenti tra loro.*
- *Il costo di acquisto dei prodotti (p) è indipendente dalla quantità ordinata, con un costo di ordinazione (g) addebitato per ogni ordine.*
- *Esiste un solo magazzino con capacità infinita.*
- *La domanda è costante e deterministica, rappresentata da D che indica la domanda annuale.*
- *Il tempo di arrivo del lotto (lead time) è nullo.*
- *La vita del prodotto è infinita.*

Il costo totale annuo è composto da tre componenti:

- *La prima componente rappresenta il costo delle materie prime, che è pari a pS.*
- *La seconda componente riguarda il costo di ordinazione, valutato come gS/q.*
- *La terza componente comprende il costo di detenzione delle scorte (il costo sostenuto per mantenere la materia prima in magazzino), che è proporzionale alla quantità media delle scorte, ossia $q/2$, con una costante m.*

Quindi, la funzione di costo totale si esprime come segue:

$$C(q) = pS + \frac{gS}{q} + \frac{mq}{2}.$$

Da cui deriva il valore ottimale di q come:

$$q^* = \sqrt{\frac{2Sg}{m}}.$$

Va notato che questo esempio è una semplificazione, poiché nel mondo reale ci sono spesso altre variabili da considerare, come la variabilità della domanda o dei tempi di consegna. Tuttavia, fornisce una base per comprendere come l'EOQ possa essere utilizzato per ottimizzare la gestione delle scorte nell'ambito della gestione degli approvvigionamenti.

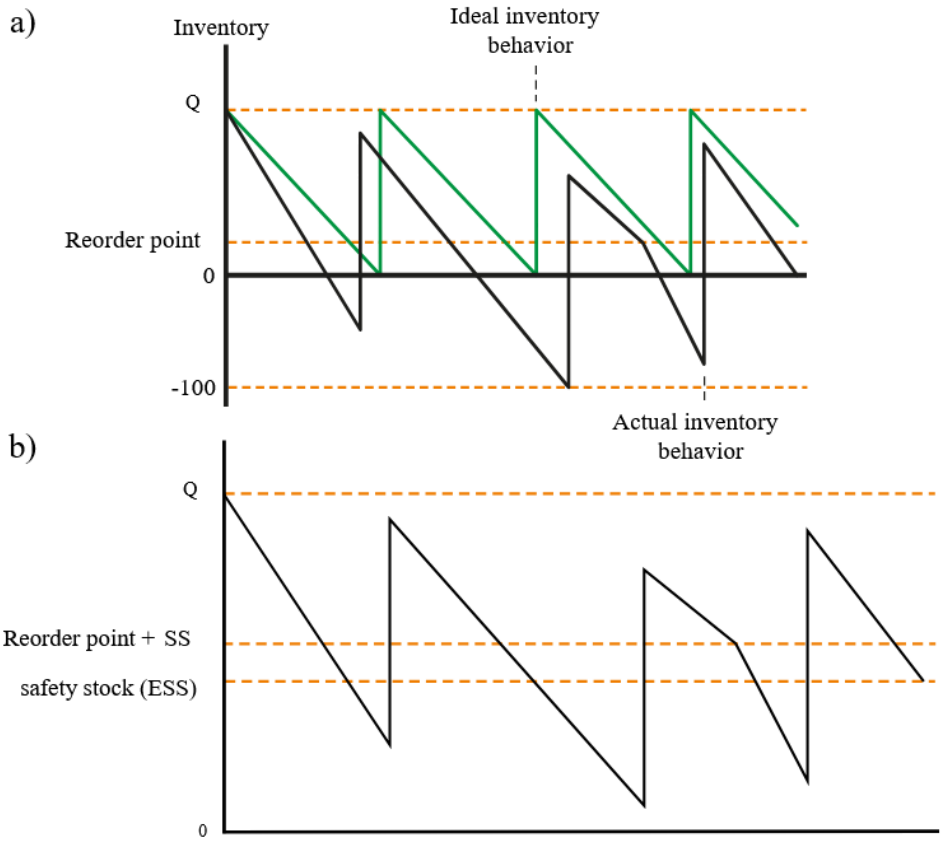

Figura 3.2: a) Visualizzazione schematica del processo di gestione delle scorte a ciclo continuo, con enfasi sul punto di riordino (ROP) per evitare carenze di inventario. b) Rappresentazione dettagliata del concetto di scorta di sicurezza nella gestione delle scorte, illustrando come questa strategia viene implementata per garantire la continuità operativa e la soddisfazione del cliente.

Gestione delle Scorte a Ciclo Continuo

Un'alternativa significativa è l'approccio di gestione delle scorte a ciclo continuo. Questo metodo si focalizza sulla costante vigilanza dell'inventario, assicurandosi che gli articoli vengano riordinati quando si raggiunge un preciso livello critico noto come "punto di riordino" (ROP). Quando le scorte scendono al di sotto del ROP, scatta un ordine di riordino. Per una comprensione più chiara di questo processo, fare riferimento all'Immagine 3.2.A. L'efficacia di questa strategia dipende dalla precisa determinazione del ROP per ciascun articolo, considerando fattori come la domanda prevista e i tempi di consegna. L'obiettivo primario è evitare carenze di inventario, garantendo al contempo la disponibilità di scorte minime per soddisfare le richieste dei clienti.

Questo approccio può essere ulteriormente affinato tramite l'uso della scorta di sicurezza, una tattica di gestione delle scorte volta a garantire la continuità e l'affidabilità delle operazioni aziendali, specialmente in situazioni impreviste. Per un'analisi più dettagliata dell'approccio con scorta di sicurezza, fare riferimento all'Immagine 3.2.A, che illustra chiaramente come questo concetto sia implementato nella gestione delle scorte. La scorta di sicurezza implica il mantenimento di un surplus di prodotti o materiali al di sopra della domanda ordinaria, a scopo precauzionale. Per calcolare la quantità necessaria di inventario di sicurezza, l'azienda prende in considerazione vari fattori, tra cui la fluttuazione nella domanda dei clienti, la variabilità nei tempi di consegna dei fornitori e il livello desiderato di servizio al cliente. L'obiettivo è determinare la quantità di inventario extra necessaria per affrontare eventuali incertezze. I vantaggi dell'Inventario di Sicurezza sono evidenti: evita situazioni in cui i prodotti non sono disponibili quando i clienti ne hanno bisogno, preservando la soddisfazione e la fedeltà del cliente. Inoltre, offre la flessibilità necessaria per adattarsi a improvvisi cambiamenti nella domanda o ritardi nella catena di approvvigionamento. Tuttavia, è essenziale evitare l'eccesso di inventario di sicurezza, poiché può comportare costi di stoccaggio elevati e il rischio di obsolescenza. La gestione efficace dell'inventario di sicurezza richiede un equilibrio accurato tra la mitigazione delle incertezze e la gestione dei costi, con la quantità di inventario di sicurezza soggetta a monitoraggio costante e adattamento in base alle mutevoli condizioni di mercato e alle esigenze aziendali.

Modello Implementato

L'applicazione a cui ho lavorato in questi mesi è stata sviluppata su misura per un'importante azienda globale operante nel settore automobilistico. La gestione dell'inventario in questo settore richiede un approccio attentamente elaborato a causa delle sfide uniche e complesse che si presentano. Una delle sfide principali è la vasta diversità di pezzi di ricambio, che spaziano dalla bulloneria ai motori. Ciascun componente richiede una strategia specifica in termini di approvvigionamento e stoccaggio, poiché la domanda e la durata in magazzino possono variare notevolmente. Inoltre, molti pezzi di ricambio destinati a veicoli fuori produzione comportano costi di produzione elevati e spesso richiedono quantità minime di ordine (MOQ) considerevoli da parte dei fornitori. La gestione dell'inventario in un'azienda globale con tempi di consegna variabili da 2 a 6 mesi rappresenta un'altra sfida significativa. In un'organizzazione di grandi dimensioni con clienti distribuiti in tutto il mondo, è fondamentale adottare una strategia di gestione dell'inventario ben strutturata e mettere in atto precise misure logistiche per garantire che i prodotti siano disponibili in tempo per soddisfare le esigenze globali dei clienti. Ciò implica non solo una pianificazione avanzata ma anche la capacità

di immagazzinare prodotti per periodi estesi senza accumuli eccessivi di inventario o problemi di obsolescenza. La gestione delle scorte a lungo termine richiede una profonda comprensione dei cicli di vita dei prodotti e delle sfide specifiche del settore automobilistico.

Come già evidenziato, le strategie tradizionali come il Just-In-Time (JIT) e la Quantità Economica d'Ordine (EOQ) potrebbero non essere facilmente adattabili alla nostra attuale situazione operativa. Questo è dovuto ai volumi significativi di inventario che gestiamo, in parte a causa dei requisiti minimi di ordine (MOQ) elevati e dei prolungati tempi di consegna associati ai nostri materiali. Pertanto, abbiamo deciso di adottare un approccio alternativo che tenga conto delle specifiche sfide che affrontiamo. La nostra scelta è stata quella di implementare una strategia di monitoraggio continuo delle scorte. Questa strategia, basata sulla costante valutazione delle nostre esigenze di rifornimento, dei punti di riordino e delle scorte di sicurezza, ci consente di gestire in modo più flessibile e reattivo, cercando un equilibrio ottimale tra l'offerta e la domanda, tenendo conto delle peculiarità delle nostre operazioni.

Un ulteriore passo significativo è stato l'approccio di categorizzazione dei nostri articoli in base alle loro caratteristiche e all'importanza strategica per il nostro processo operativo. Abbiamo creato diverse matrici logistiche per classificare i pezzi in categorie che rispecchiano la loro applicabilità e rilevanza per il nostro funzionamento. Ad esempio, la bulloneria, essenziale per qualsiasi operazione di manutenzione, è stata collocata in una categoria con scorte più abbondanti per garantire una disponibilità costante. D'altra parte, articoli costosi come i motori, per i quali i clienti sono disposti ad attendere, sono stati categorizzati in modo da minimizzare il rischio di avere invenduti in magazzino, tenendo conto non solo dei costi di mantenimento delle scorte, ma anche delle implicazioni finanziarie legate all'accumulo di articoli costosi non venduti, che comportano costi aggiuntivi oltre al valore dell'inventario stesso.

3.1.1 Metodi per il Calcolo delle Scorte di Sicurezza

Esaminiamo ora approfonditamente la nostra strategia di monitoraggio continuo e come questa incida sul nostro punto di riordino, il safety stock, il livello di servizio e il fill rate. Attraverso questa analisi, otteniamo una visione completa della gestione delle nostre risorse e della capacità di soddisfare le esigenze dei nostri clienti. Inoltre, sfruttando algoritmi di previsione sofisticati, come quelli delineati nel capitolo precedente, siamo in grado di calcolare con precisione la quantità di prodotto da produrre per coprire la prevista domanda di vendita. Il concetto di safety stock diventa fondamentale quando sussiste incertezza nelle previsioni della domanda o nei tempi di consegna dei prodotti, agendo come un'assicurazione

contro potenziali carenze di scorte. La quantità di safety stock necessaria è direttamente proporzionale alla precisione delle previsioni: maggiore è l'incertezza, maggiore sarà la necessità di una scorta di sicurezza per garantire un determinato livello di servizio.

La determinazione del safety stock sulla base del livello di servizio e del fill rate rappresenta due approcci distinti nella gestione delle scorte, ciascuno con una prospettiva diversa sull'obiettivo principale e sull'approccio strategico. Il livello di servizio è principalmente orientato al cliente ed è comunemente espresso come la probabilità di soddisfare la domanda dei clienti D durante un ciclo di ordine senza incorrere in esaurimenti delle scorte. La sua formula è la seguente:

$$\text{Livello di Servizio} = 1 - \mathbb{P}(D > ROP) = F_D(ROP)$$

Quando si calcola il safety stock basandosi su questa metrica, l'obiettivo principale è garantire che una percentuale specifica delle richieste dei clienti venga soddisfatta in modo affidabile, senza incorrere in esaurimenti delle scorte. Questo approccio è proattivo e tiene conto della variabilità della domanda, dei tempi di consegna e del livello desiderato di servizio. È incentrato sulla prevenzione di situazioni in cui i clienti potrebbero affrontare scorte insufficienti, mantenendo un adeguato margine di sicurezza.

L'approccio generale utilizzato in questo contesto per calcolare le scorte di sicurezza si basa su diversi fattori chiave:

- La domanda, che rappresenta il numero di articoli effettivamente consumati dai clienti.
- L'errore di previsione, che stima quanto la domanda effettiva possa discostarsi dalla previsione iniziale.
- Il tempo di attesa, che indica il ritardo tra il momento in cui si raggiunge il punto di riordino (ROP) e la disponibilità rinnovata delle scorte.
- Il livello di servizio, che rappresenta la probabilità desiderata di soddisfare la domanda durante il tempo di attesa senza incorrere in esaurimenti delle scorte. Aumentando il livello di servizio, aumenta anche la quantità di scorta di sicurezza richiesta.

Assumendo che la domanda durante i periodi successivi sia costituita da variabili casuali indipendenti e distribuite secondo una distribuzione normale, il calcolo della scorta di sicurezza può essere eseguito utilizzando la seguente formula:

$$SS = z_\alpha \times \sqrt{\mathbb{E}(L)\sigma_D^2 + (\mathbb{E}(D))^2\sigma_L^2},$$

Dove:

- α rappresenta il livello di servizio desiderato, mentre z_α è il quantile di ordine α di una distribuzione normale standard.
- $\mathbb{E}(L)$ e σ_L sono rispettivamente la media e la deviazione standard del tempo di attesa.
- $\mathbb{E}(D)$ e σ_D rappresentano la media e la deviazione standard della domanda per ciascun periodo di tempo.

Il punto di riordino (ROP) può quindi essere calcolato sommando la domanda media durante un ciclo di ordine, con la scorta di sicurezza

$$ROP = \mathbb{E}(L) \cdot \mathbb{E}(D) + SS.$$

D'altro canto, il fill rate è un indicatore che si concentra sull'efficienza operativa. Quando si calcola il safety stock basandosi su questa metrica, l'attenzione principale è rivolta all'esecuzione degli ordini correnti e all'ottimizzazione dei processi di gestione delle scorte. Il fill rate misura la percentuale di ordini dei clienti che sono stati completati senza problemi o ritardi nelle consegne. La sua formula è piuttosto semplice:

$$\text{Fill Rate} = \frac{\text{Numero di Ordini Completi}}{\text{Numero Totale di Ordini}}$$

Seguendo l'approccio di Nahmias and Olsen [2015], definiamo il numero atteso di esaurimenti di scorte $L(z)$, ovvero il numero medio di volte in cui un prodotto rimarrà senza scorte entro un determinato periodo. La formula per $L(z)$ è la seguente:

$$L(z) = \int_z^{+\infty} (t - z)\, \phi(t) dt$$

Dove $\phi(t)$ è la funzione di densità della distribuzione normale standard. Considerando β come il fill rate minimo desiderato, Q la quantità ordinata e σ la deviazione standard della domanda, possiamo stabilire la seguente relazione:

$$L(z) = (1 - \beta)\frac{Q}{\sigma}$$

In questo contesto, la scorta di sicurezza può essere calcolata come segue:

$$SS = z_\beta \sigma$$

La scelta tra il calcolo del safety stock basato sul livello di servizio o basato sul fill rate dipende dalle priorità e dagli obiettivi specifici dell'azienda. Se l'azienda attribuisce la massima importanza alla soddisfazione del cliente e alla prevenzione degli esaurimenti delle scorte, allora il calcolo basato sul livello di servizio è più

appropriato. Al contrario, se l'azienda mira a ottimizzare l'efficienza operativa e l'elaborazione degli ordini, allora il calcolo basato sul fill rate potrebbe essere la scelta preferibile. Nel contesto dei magazzini che stiamo analizzando, le richieste di approvvigionamento non provengono direttamente dai clienti finali, ma piuttosto attraverso gli ordini dei rivenditori. Pertanto, dal punto di vista dell'azienda, è più logico adottare una prospettiva orientata all'ottimizzazione dei propri processi interni. In questa situazione, il calcolo dell'inventario di sicurezza e del punto di riordino è spesso basato sul concetto di fill rate.

3.2 Fitting della domada

Fino a questo punto, abbiamo esaminato come calcolare il livello di sicurezza (safety stock) e il punto di riordino dati un livello di servizio o un fill rate target, ma abbiamo sempre operato con l'assunzione che la domanda seguisse una distribuzione normale. Ora, vogliamo approfondire e ampliare la nostra analisi considerando le distribuzioni appropriate per il comportamento della domanda.

La domanda di pezzi di ricambio è notoriamente caratterizzata da un comportamento estremamente variabile, dovuto alla sua natura intrinseca. La maggior parte della domanda di pezzi di ricambio è causata dai guasti di apparecchiature meccaniche o elettroniche, il che la rende altamente imprevedibile. Questa imprevedibilità suggerisce che dovremmo utilizzare distribuzioni composte, poiché queste ci permettono di modellare le dimensioni delle richieste e gli intervalli tra di esse come variabili distinte. Nella letteratura sulla domanda intermittente, i tempi tra le richieste sono stati modellati sia con variabili discrete che continue.

Quando trattiamo il tempo come una variabile discreta, l'arrivo delle richieste viene comunemente modellato come un processo di Bernoulli. In altre parole, in ogni periodo, la probabilità di ricevere una richiesta è rappresentata da una variabile casuale di Bernoulli, e i tempi tra le richieste seguono una distribuzione geometrica. La quantità richiesta rappresenta la somma di tutte le richieste che arrivano durante un periodo di tempo specifico. Un modello introdotto da Croston utilizza questa struttura composita per prevedere la domanda intermittente, in cui l'arrivo delle richieste è descritto da un processo di Bernoulli e le quantità richieste sono modellate come variabili casuali normali indipendenti e identicamente distribuite. Tuttavia, secondo Syntetos et al. [2011], l'uso della distribuzione normale per modellare le quantità richieste ha poco supporto empirico nel contesto dei pezzi di ricambio, in quanto la domanda intermittente tende ad essere fortemente asimmetrica a destra, rendendo la normalità un'assunzione inappropriata. Altre distribuzioni composte basate su Bernoulli sono state proposte, come la distribuzione logaritmico-Poisson per le quantità richieste, ma sono state superate in termini di performance dalla distribuzione binomiale negativa (NBD).

Quando modelliamo il tempo come una variabile continua, l'arrivo delle richieste è solitamente descritto da un processo di Poisson, con intervalli tra le richieste distribuiti in modo esponenziale. In questo caso, ogni richiesta è considerata indipendentemente, e se modelliamo le quantità richieste come variabili geometriche, otteniamo una distribuzione binomiale negativa delle richieste per unità di tempo.

Nonostante l'adeguatezza delle distribuzioni composte per modellare la domanda intermittente, esistono alcune distribuzioni non composte che sono state ampiamente utilizzate. In particolare, quando si hanno tempi di attesa molto lunghi, si può fare affidamento sul teorema del limite centrale, il che rende la distribuzione normale un'assunzione ragionevole per modellare la domanda in attesa. La distribuzione normale è spesso utilizzata come riferimento nelle applicazioni pratiche e viene confrontata con altre distribuzioni più complesse. Inoltre, la distribuzione gamma è comunemente utilizzata quando si tratta di modellare la domanda in attesa, soprattutto grazie alla sua tracciabilità analitica e alla sua capacità di rappresentare solo valori positivi. In sintesi, abbiamo esaminato varie opzioni per modellare la domanda di pezzi di ricambio, considerando sia distribuzioni composte come la NBD, sia distribuzioni non composte come la distribuzione normale, la gamma e la Poisson. La scelta dipenderà dalle specifiche caratteristiche del caso di studio e dalla bontà dell'adattamento empirico dei dati.

Test di Kolmogorov-Smirnov

Il test di Kolmogorov-Smirnov è uno strumento statistico fondamentale impiegato per determinare se un insieme di dati segue una distribuzione di probabilità specifica o presenta somiglianze con una distribuzione teorica conosciuta, che rappresenta l'ipotesi nulla. Questo test è ampiamente utilizzato per valutare se un campione di dati è estratto da una popolazione che segue una distribuzione nota, come ad esempio la distribuzione normale o un'altra distribuzione specifica.

Per comprendere il funzionamento di questo test, è essenziale considerare la rappresentazione dei dati osservati mediante la Funzione di Distribuzione Cumulativa Empirica (ECDF). La ECDF è una funzione che assegna a ciascun valore osservato la probabilità che una variabile casuale estratta dalla stessa popolazione sia minore o uguale a quel valore. In termini matematici, la ECDF visualizza quanto i dati osservati si allineino con la distribuzione teorica.

La formula della ECDF per un valore x è la seguente:

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} I(x_i \leq x)$$

Dove:

- $\hat{F}_n(x)$ rappresenta la ECDF per x.

- n è il numero totale di dati nel campione.
- x_i sono i valori osservati nel campione.
- $I(x_i \leq x)$ è una funzione indicatrice che assume il valore 1 se x_i è minore o uguale a x, altrimenti assume il valore 0.

In pratica, la ECDF di x indica la proporzione di dati nel campione che sono minori o uguali a x. Questa rappresentazione empirica della distribuzione dei dati fornisce una stima visuale della distribuzione di probabilità basata sui dati raccolti.

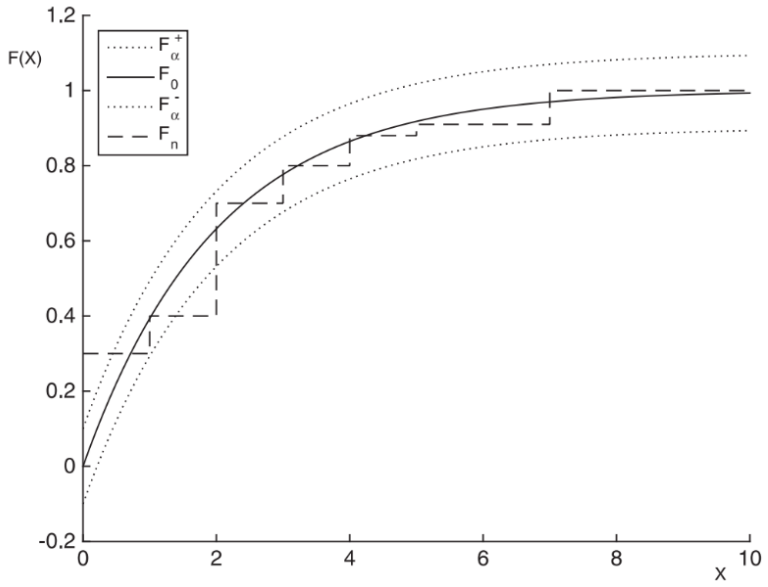

Figura 3.3: Rappresentazione del Test di Kolmogorov-Smirnov

La statistica D nel test di Kolmogorov-Smirnov è definita come la massima discrepanza tra la ECDF dei dati osservati ($F_n(x)$) e la Funzione di Distribuzione Cumulativa (CDF) della distribuzione teorica specificata ($F(x)$). Per calcolare D, si determina la differenza $D(x)$ per ogni valore x:

$$D(x) = |F_n(x) - F(x)|$$

Successivamente, si individua il valore massimo di $D(x)$ su tutto l'intervallo dei dati osservati:

$$D = \max_x D(x)$$

Questo valore D rappresenta la massima discrepanza tra i dati osservati e la distribuzione teorica ipotizzata ed è utilizzato per prendere una decisione riguardo all'accettazione o al rifiuto dell'ipotesi nulla nel test di Kolmogorov-Smirnov.

```
def apply_ks_test(row, n, distribution, args):
    n = int(n)
    statistic, p_value = kstest(row[:n], distribution, args=args)
    return pd.Series({'Statistic_' + distribution: statistic,
        'p-value_' + distribution: p_value})

norm_results = data[qty + ['n', 'mean_qty',
    'var_qty']].apply(lambda row: apply_ks_test(row[:-3], row[-3],
    'norm', args=(row[-2], row[-1] ** 0.5)), axis=1)
data[['Statistic_norm', 'p-value_norm']] = norm_results

poisson_results = data[qty + ['n', 'mean_qty',
    'var_qty']].apply(lambda row: apply_ks_test(row[:-3], row[-3],
    'poisson', args=(row[-2])), axis=1)
data[['Statistic_poisson', 'p-value_poisson']] = poisson_results

gamma_results = data[qty + ['n', 'mean_qty',
    'var_qty']].apply(lambda row: apply_ks_test(row[:-3], row[-3],
    'gamma', args=((row[-2]*row[-2]/row[-1], 0,
    row[-2]/row[-1]))), axis=1)
data[['Statistic_gamma', 'p-value_gamma']] = gamma_results

nbinom_results = data[qty + ['n', 'mean_qty',
    'var_qty']].apply(lambda row: apply_ks_test(row[:-3], row[-3],
    'nbinom', args=(row[-2]*row[-2]/(row[-1] - row[-2]),
    row[-2]/row[-1])), axis=1)
data[['Statistic_nbinom', 'p-value_nbinom']] = nbinom_results
```

Codice 3.1: Algoritmo che valutano la statistica ed il p-value del KS-Test per 4 distrubuzioni.

Nel 3.1, viene presentato il codice implementato per eseguire il test di Kolmogorov-Smirnov su un dataset composto da 100.000 pezzi. Basandoci sulla letteratura introdotta all'inizio di questa sezione, abbiamo testato diverse distribuzioni, tra cui la distribuzione normale, gamma, Poisson e binomiale negativa. Per stimare queste distribuzioni, abbiamo prima calcolato direttamente due momenti statistici, ovvero la media campionaria $\hat{\mu}$ e la varianza campionaria $\hat{\sigma^2}$, utilizzando i dati osservati sulla domanda. Questi momenti sono stati successivamente impiegati come parametri diretti per le distribuzioni normale e Poisson, dove abbiamo assegnato $\lambda = \hat{\mu}$ per la distribuzione Poisson. Per quanto riguarda la distribuzione gamma, i parametri sono stati stimati mediante le seguenti formule:

$$a = \frac{\hat{\mu}^2}{\hat{\sigma^2}}$$

$$b = \frac{\hat{\sigma^2}}{\hat{\mu}}$$

Per le distribuzioni binomiali negative, abbiamo utilizzato i seguenti calcoli per stimare i parametri:

$$p = 1 - \frac{\hat{\mu}}{\hat{\sigma^2}}$$

$$r = \frac{\hat{\mu}(1-p)}{p}$$

Dopo aver calcolato la statistica D tramite il test KS, abbiamo proceduto al calcolo del p-value, il quale ci ha permesso di decidere se accettare o respingere l'ipotesi nulla del test. Seguendo il metodo di riferimento Turrini and Meissner [2019], abbiamo considerato una corrispondenza "forte" se la statistica D rientrava nell'intervallo di confidenza al 95%, mentre abbiamo definito una corrispondenza "buona" se rientrava nell'intervallo di confidenza al 99%.

La tabella 3.1 riepiloga i risultati ottenuti, evidenziando che circa l'80% dei pezzi ha mostrato una buona aderenza con almeno una delle distribuzioni di domanda testate. Tuttavia, è interessante notare che solo una piccola minoranza di pezzi ha ottenuto un forte adattamento con la distribuzione gamma (107 casi) e un buon adattamento (239 casi). In seguito a questa analisi, in collaborazione con il cliente, si è deciso di escludere la distribuzione gamma dall'adozione sistematica, in quanto i costi computazionali necessari per l'implementazione non giustificano l'uso generalizzato. Nella Figura 3.4, è possibile osservare una rappresentazione grafica della distribuzione dei vari cluster in funzione dell'intervallo medio tra le domande e del coefficiente di variazione al quadrato. In particolare, si nota che i cluster sono approssimativamente separabili.

Tabella 3.1: Distribuzioni Forti e Buone

Tipo di Distribuzione	Numero
Forte Normale	32072
Forte Poisson	7631
Forte Gamma	107
Forte Binomiale Negativa	35040
Buona Normale	16949
Buona Poisson	10548
Buona Gamma	239
Buona Binomiale Negativa	8899

Dopo aver condotto l'analisi, ci siamo confrontati con il problema delle prestazioni computazionali. L'applicazione web sviluppata doveva analizzare gli indicatori di magazzino relativi a decine di milioni di pezzi di domanda, e l'approccio

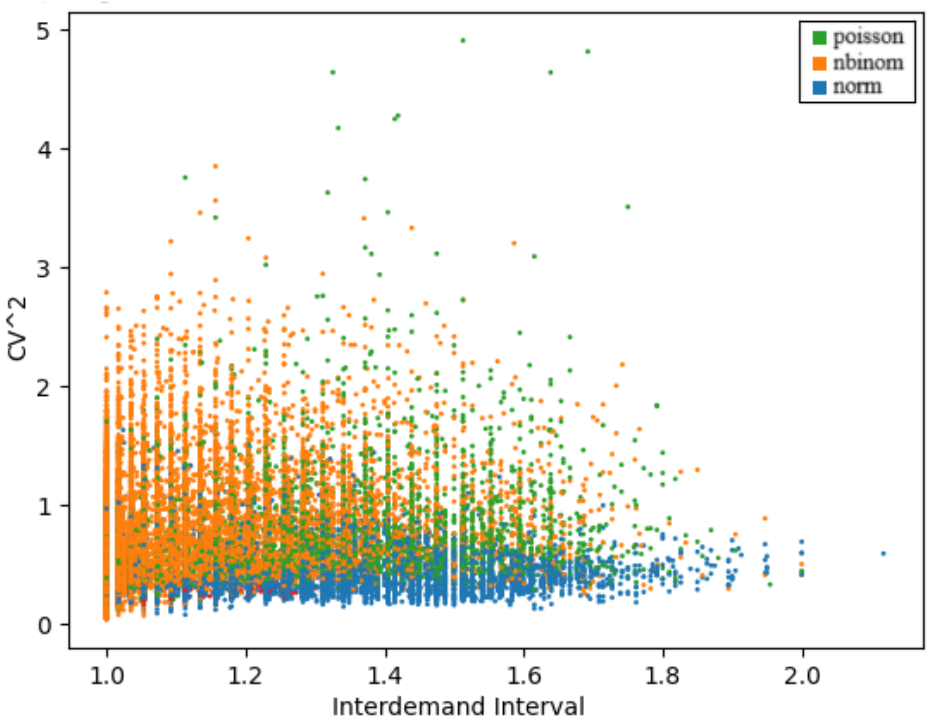

Figura 3.4: Struttura Gerarchica dei Magazzini

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_text
from sklearn.metrics import accuracy_score

feature = ['cv2', 'mean_qty', 'var_qty', 'inter_demand_interval']
clf1 = DecisionTreeClassifier(max_depth=2,
    class_weight='balanced')
clf1.fit(train_data[feature], train_data['label'])

y_pred = clf1.predict(test_data[feature])
accuracy = accuracy_score(test_data['label'], y_pred)
```

Codice 3.2: Algoritmo che valuta la statistica ed il p-value del KS-Test per 4 distrubuzioni.

di eseguire il test di Kolmogorov-Smirnov per ciascun pezzo si è rivelato impraticabile, richiedendo giorni per essere completato. Pertanto, abbiamo cercato una regola empirica che ci permettesse di decidere quando utilizzare una distribuzione rispetto a un'altra. In collaborazione con il cliente, abbiamo optato per una regola semplice e facilmente interpretabile, basata su indicatori chiave come la media, la

varianza, l'intervallo medio delle domande e il coefficiente di variazione. Questa scelta è stata guidata anche dalla volontà di mantenere la flessibilità per futuri sviluppi, come l'aggiunta di ulteriori distribuzioni di domanda.

La soluzione che abbiamo adottato è stata quella di utilizzare un albero di classificazione. Abbiamo diviso il dataset in training e validation, associando a ciascun pezzo la sua distribuzione di domanda corrispondente. Abbiamo poi eseguito numerosi test per determinare quale configurazione di albero di classificazione avesse le migliori performance. Nel 3.2, presentiamo il codice relativo alle fasi di addestramento e validazione dell'algoritmo che ha dimostrato di ottenere le prestazioni più elevate. L'albero di classificazione risultante ha la seguente forma

```
Se mean_qty ≤ 5.32 :
  Se cv2 ≤ 0.41 :
    Classe: norm
  Altrimenti:
    Classe: poisson
Altrimenti:
  Se inter_demand_interval ≤ 1.25 :
    Classe: nbinom
  Altrimenti:
    Classe: norm
```

e offre un'accuratezza di circa l'80% sul dataset di validazione, garantendo una solida base per l'applicazione pratica di questa strategia decisionale.

3.3 Organizzazione Logistica dei Magazzini

Nel mondo aziendale moderno, la gestione efficace dei magazzini è una componente fondamentale per il successo in un ambiente globale e complesso. La distribuzione di fornitori in diverse regioni del mondo e la presenza di magazzini in molteplici località richiedono un approccio strategico e organizzato per gestire le scorte e garantire un flusso ottimale nella catena di approvvigionamento. All'interno dell'applicazione aziendale su cui ho lavorato, ci siamo trovati di fronte a una rete estremamente complessa e diversificata di fornitori e una serie di magazzini di centrali e periferici dislocati in numerose regioni del globo. In questa rete, spesso è presente una struttura gerarchica tra i magazzini, con diversi livelli di funzionalità e responsabilità. Questa gerarchia è illustrata chiaramente nella figura 3.5.

Un elemento chiave di questa struttura gerarchica è il ruolo centrale del magazzino di primo livello. Questo magazzino rappresenta il punto principale di ingresso

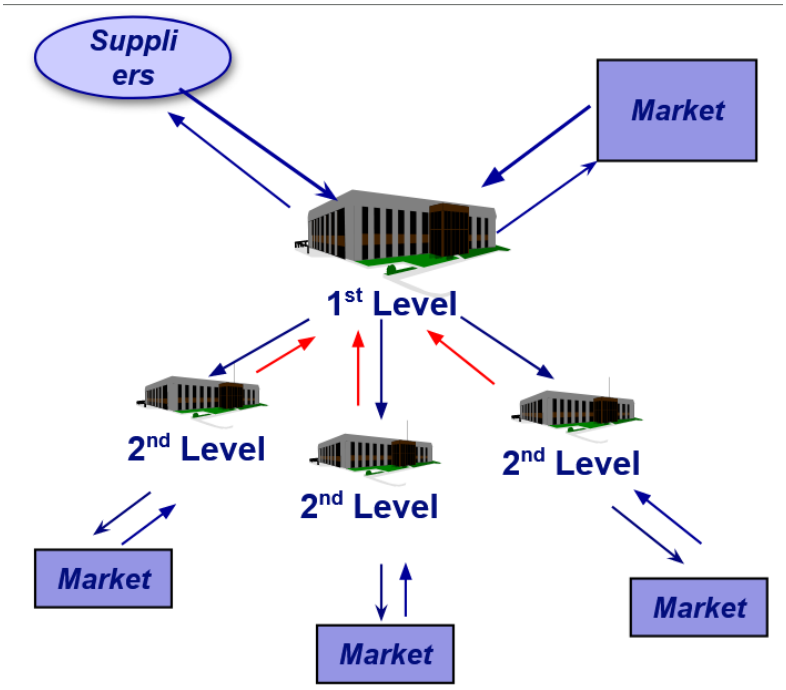

Figura 3.5: Struttura Gerarchica dei Magazzini

per le forniture provenienti dai fornitori ed è generalmente il deposito principale che mantiene le scorte più ampie. Tuttavia, l'azienda ha anche la possibilità di configurare il magazzino di primo livello come un "magazzino senza scorte", il cui ruolo principale è ricevere e distribuire le forniture in arrivo senza accumularle in grandi quantità. Parallelamente, il magazzino di secondo livello svolge un ruolo cruciale nell'approvvigionare i mercati locali e interagisce con il magazzino centrale attraverso processi complessi come la gestione degli ordini in backorder e l'allocazione automatica delle scorte.

Nel proseguo del capitolo, esamineremo più dettagliatamente le variazioni possibili all'interno di questa struttura gerarchica di magazzini, come rappresentato nella figura 3.6. Ogni variante offre vantaggi specifici, dai benefici della centralizzazione del controllo alla flessibilità della distribuzione mirata. Ogni decisione riguardo alla configurazione della catena di approvvigionamento influenzerà direttamente la gestione delle scorte, i tempi di consegna e i costi operativi. La comprensione di queste opzioni è fondamentale per adattare con successo la catena di approvvigionamento alle esigenze specifiche dell'azienda.

Ci sono quattro strutture gerarchiche fondamentali per l'organizzazione logistica dei magazzini:

- Modello Gerarchico LVL1 - LVL2: In questo scenario, il magazzino di primo

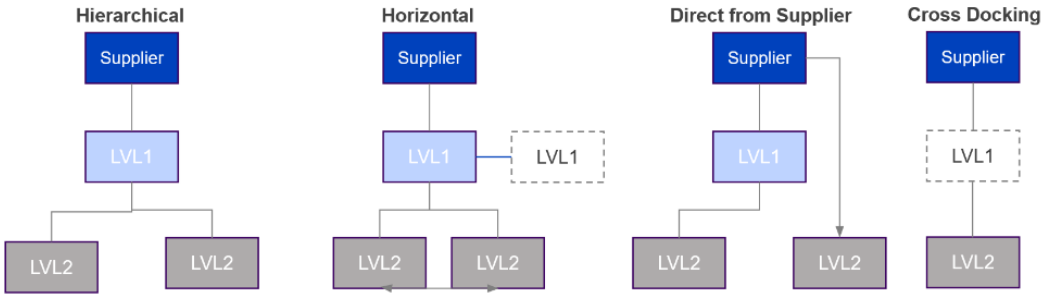

Figura 3.6: Panoramica delle diverse strutture gerarchiche tra i magazzini.

livello (LVL1) assume una posizione centrale. Funge da punto di ingresso principale per le forniture provenienti dai fornitori, gestendo il più grande stock di merci nell'intera catena di approvvigionamento. Il suo ruolo fondamentale è quello di consolidare e distribuire le merci verso i magazzini di secondo livello (LVL2). Questi ultimi sono le destinazioni finali per le merci, ma di solito non condividono le loro scorte con altri magazzini di secondo livello. Questa struttura è particolarmente vantaggiosa quando si desidera un controllo centralizzato sulla distribuzione.

- Modello Orizzontale: Nel modello orizzontale, i magazzini di secondo livello continuano a ricevere forniture dai magazzini di primo livello. La novità sta nella loro flessibilità nel poter scambiare scorte tra di loro, se necessario. Questo approccio favorisce una distribuzione più dinamica e consente una risposta rapida alle esigenze dei mercati locali. Inoltre, i magazzini di primo livello possono scambiare scorte con altri magazzini di primo livello situati in diverse regioni, se questa strategia è rilevante per l'azienda..

- Modello Direct from Suppliers: Il modello "Direct from Suppliers" semplifica il flusso di approvvigionamento. Qui, i magazzini di secondo livello possono ricevere forniture direttamente dai fornitori, bypassando completamente il magazzino di primo livello. Questo è particolarmente vantaggioso quando si desidera accelerare la distribuzione e ridurre i passaggi intermedi nella catena di approvvigionamento.

- Modello Cross Docking: Il modello di "Cross Docking" rappresenta un approccio altamente efficiente alla distribuzione. Quando le forniture arrivano dai fornitori, il magazzino di primo livello non accumula le merci ma le inoltra direttamente ai magazzini di secondo livello. Questo riduce al minimo i tempi di stoccaggio e i costi associati. È particolarmente utile quando è essenziale consegnare rapidamente le merci ai clienti senza accumularle nei magazzini di primo livello.

In sintesi, queste varianti nella struttura gerarchica dei magazzini rappresentano una risorsa preziosa per l'azienda, permettendo loro di adattare la catena di approvvigionamento alle specifiche esigenze del mercato. Questa adattabilità si traduce in un'ottimizzazione sia nella distribuzione che nella gestione delle scorte, con notevoli vantaggi in termini di tempi di consegna, controllo centralizzato e costi di trasporto. A seconda del conteto, la scelta della struttura più adeguata dipenderà dalla strategia aziendale e dalle dinamiche specifiche della catena di approvvigionamento.

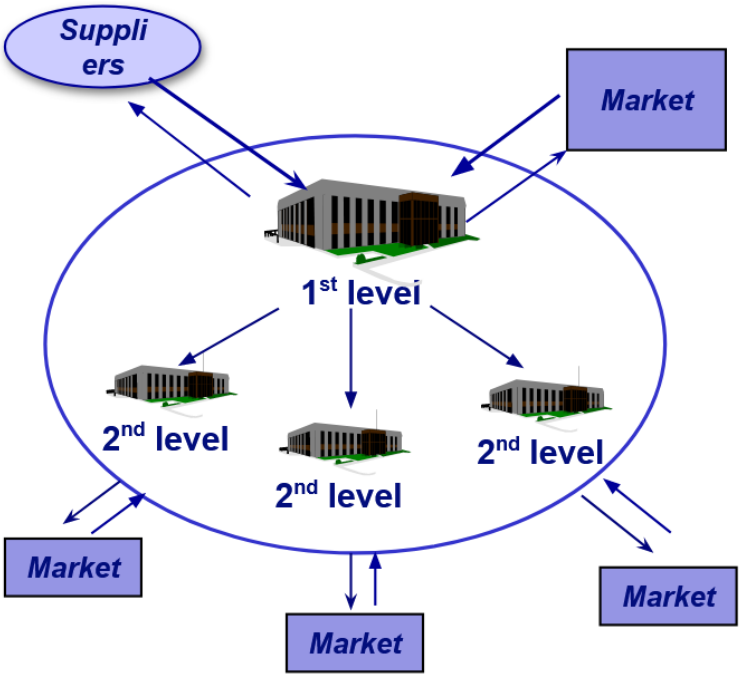

Figura 3.7: Aggregazione dei magazzini di primo e di secondo livello in un magazzino virtuale.

Introducioamo ora il concetto di Bill of Distribution (BOD), questo concetto costituisce un pilastro fondamentale nella pianificazione della distribuzione e definisce le vie attraverso cui i prodotti viaggiano all'interno dell'azienda. La Bill of Distribution specifica il percorso di distribuzione di un prodotto all'interno dell'azienda, dal fornitore al cliente finale. Grazie alla definizione chiara delle vie di distribuzione fornita dalla BOD, diventa possibile pianificare in modo efficiente le parti di servizio, evitando processi di determinazione delle fonti di approvvigionamento lunghi e complessi. Nell'applicazione che abbiamo sviluppato, è possibile creare, modificare, eliminare o visualizzare le BOD. Le BOD ci hanno permesso, inoltre, di introdurre il concetto di "magazzino virtuale," dove le scorte di tutti i magazzini correlati, inclusi quelli di primo e secondo livello, possono essere gestite

come se facessero parte di un unico "Magazzino Virtuale." Questo approccio, illustrato nella figura 3.7, offre una visione aggregata delle scorte aziendali, facilitando ulteriormente la gestione e l'ottimizzazione delle risorse.

Il concetto di warehouse virtuale si presenta come uno strumento fortemente vantaggioso. Inizialmente, contribuisce in modo significativo a migliorare l'accuratezza delle previsioni di domanda. Riunendo la domanda proveniente da tutti i magazzini inclusi nella stessa Bill of Distribution (BOD), appartenenti quindi al magazzino virtuale, si ottiene una domanda complessiva caratterizzata da una minore incertezza. Questa riduzione dell'incertezza apre la strada all'applicazione di algoritmi di previsione, come descritti nel capitolo 2, i quali possono essere addestrati su dati più stabili, portando così a previsioni più precise. Inoltre, l'associazione della distribuzione di domanda, come analizzata nella sezione precedente 3.2, risulta essere più efficace, migliorando ulteriormente la precisione delle previsioni, elemento di cruciale importanza per una gestione ottimale del magazzino.

3.3.1 Conclusione

L'algoritmo finale implementato per la gestione dell'inventario si articola in diverse fasi fondamentali. Inizialmente, viene eseguita un'analisi per ciascun pezzo di ricambio in ogni magazzino, indipendentemente dalla struttura logistica di appartenenza. Durante questa fase, vengono effettuate stime della domanda futura $\hat{\mu}$ e della sua variabilità $\hat{\sigma^2}$ utilizzando gli algoritmi specifici di previsione. Queste stime costituiscono la base per adattare la distribuzione parametrica della domanda, la quale viene successivamente utilizzata per calcolare indicatori di magazzino fondamentali, tra cui safety stock e punto di riordino. Questo processo di previsione e calcolo degli indicatori di magazzino viene applicato anche a livello di magazzino virtuale per ciascun pezzo di ricambio.

L'ultimo passo cruciale in questo algoritmo è rappresentato dalla fase di quadratura, il cui procedimento è illustrato nella Figura 3.8. In questa fase, la classe di movimentazione (fast e slow mover) dei prodotti riveste un ruolo chiave. Se i valori di sicurezza delle scorte e i punti di riordino risultano essere più elevati a livello virtuale rispetto alla somma dei magazzini fisici, ciò suggerisce che le stime di vendita per i singoli magazzini sono sottostimate. In questa situazione, per i prodotti ad alta movimentazione, si tende ad aumentare le scorte nei magazzini di secondo livello, poiché sono quelli che ricevono la maggior parte degli ordini dai clienti. Questo incremento di scorte ha lo scopo di soddisfare la domanda aggiuntiva nel breve termine, incrementando, di conseguenza, il fill rate. Per quanto riguarda i prodotti a movimentazione più lenta, al contrario, si tende ad accumulare le scorte aggiuntive nel magazzino di primo livello. Questa strategia facilita il loro trasporto verso il magazzino appropriato quando si verifica effettivamente la

domanda. È importante ricordare che gli slow mover sono prodotti per i quali è difficile prevedere il momento e il luogo dell'ordine.

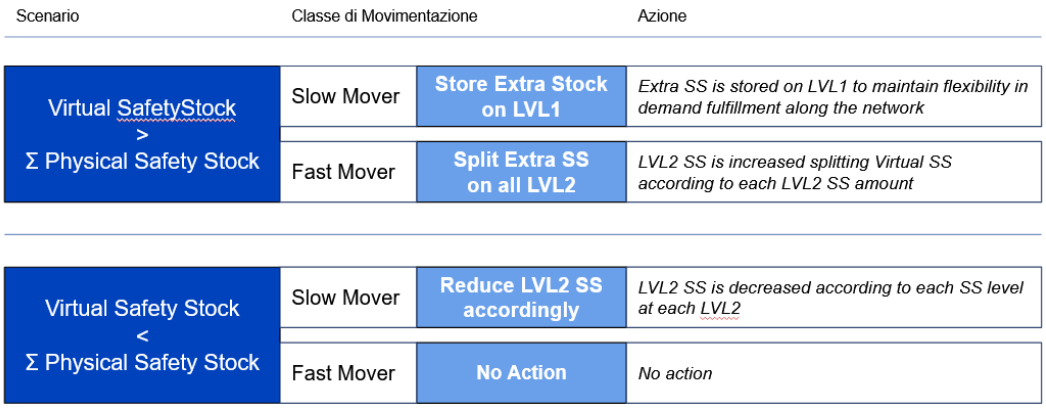

Figura 3.8: Riassunto delle politiche di gestione della scorta di sicurezza.

Invece, se i valori di sicurezza delle scorte e i punti di riordino risultano essere più bassi a livello virtuale rispetto alla somma dei magazzini fisici, ciò indica una sovrastima delle vendite per i singoli magazzini. In questo scenario, per i prodotti ad alta movimentazione, non viene effettuata una riduzione dei valori di scorta. Questi articoli sono meno soggetti al rischio di obsolescenza, quindi l'extra stock presente in questi magazzini è generalmente destinato a essere venduto, evitando così costi aggiuntivi di trasporto. Per quanto concerne i prodotti a movimentazione più lenta, si procede a una riduzione delle scorte principalmente nel magazzino di secondo livello, seguendo quanto descritto in precedenza. Le scorte vengono quindi accumulate nel magazzino di primo livello per migliorare la gestione logistica dei pezzi.

Ringraziamenti

Mi è doveroso dedicare questo spazio alle persone che hanno contribuito, con il loro supporto, alla realizzazione di questo elaborato.
Un ringraziamento speciale al mio relatore Prof. Brandimarte, per i suoi indispensabili consigli e per le conoscenze trasmesse durante tutto il percorso.
Ringrazio infinitamente i miei genitori che mi hanno sempre sostenuto, appoggiando ogni mia decisione e aiutandomi a raggiungere tutti i miei obiettivi.
Un grazie di cuore a Monica, che mi ha accompagnato nell'intero percorso universitario. È grazie a lei che ho superato i momenti più difficili.
Infine, dedico questa tesi a me stesso, ai miei sacrifici e alla mia tenacia che mi hanno permesso di arrivare fin qui.

Bibliografia

Chung Chen and Lon-Mu Liu. Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association*, 88(421): 284–297, 1992. doi: 10.2307/2290724.

R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(3):1–22, 2008. doi: 10.18637/jss.v027.i03.

S. Nahmias and T. L. Olsen. *Production and Operations Analysis*. Waveland Press, 2015.

A A Syntetos, J E Boylan, and J D Croston. On the categorization of demand patterns. *Journal of the Operational Research Society*, 56(5):495–503, 2005. doi: 10.1057/palgrave.jors.2601841.

Aris Syntetos and John Boylan. The accuracy of intermittent demand estimates. *International Journal of Forecasting*, 21(2):303–314, 2005. doi: 10.1016/j.ijforecast.2004.10.001.

Aris Syntetos, Mohamed Zied Babai, and Nezih Altay. On the demand distributions of spare parts. *International Journal of Production Research*, 50(8): 2101–2117, 2012. doi: 10.1080/00207543.2011.562561.

Aris A. Syntetos, M. Zied Babai, David Lengu, and Nezih Altay. *Distributional Assumptions for Parametric Forecasting of Intermittent Demand*, pages 31–52. Springer London, 2011. doi: 10.1007/978-0-85729-039-7_2.

Laura Turrini and Joern Meissner. Spare parts inventory management: New evidence from distribution fitting. *European Journal of Operational Research*, 273 (1):118–130, 2019. doi: https://doi.org/10.1016/j.ejor.2017.09.039.