



Dipartimento di Ingegneria Gestionale e della Produzione
Corso di Laurea Magistrale in INGEGNERIA GESTIONALE
A.a. 2024/2025
Sessione di Laurea Ottobre 2025

Tesi di Laurea
Ottimizzazione della Gestione delle Scorte e del
Riordino: Il Caso IGLM STORE

Relatore:

Prof. Eliana Pastor

Candidato:

Antonino Pollio

Tutor Aziendale:

Gaetano Incarnato

Abstract

La gestione delle scorte rappresenta una delle sfide più critiche per le imprese e-commerce multicanale, dove la rapidità di consegna, la disponibilità costante dei prodotti e la corretta distribuzione tra diversi canali di vendita costituiscono fattori determinanti per la competitività. La presente tesi analizza il caso di IGLM.Store, impresa italiana con un catalogo multi settoriale e canali distributivi diversi (eCommerce diretto, Amazon FBM e Amazon FBA). L'azienda è coinvolta nella gestione di problematiche tipiche dei contesti complessi, quali rotture di stock, accumuli di merce invenduta e trasferimenti non ottimali di prodotti verso i magazzini Amazon.

Per rispondere a tali criticità, la ricerca propone lo sviluppo di un sistema avanzato di Business Intelligence e Data Analytics basato su un'architettura modulare, capace di integrare dati interni (storico vendite, giacenze, lead time) inviati dal gestionale aziendale con dati esterni provenienti dal marketplace Amazon (Buy Box, ranking dei prodotti, prezzi e competitività) e recuperati automaticamente tramite API. Il sistema possiede due elementi essenziali: un algoritmo di classificazione, che per ogni prodotto raccomanda l'operazione più adeguata (riordino dal fornitore, trasferimento a FBA o nessuna operazione), ed un modello SARIMA di previsione, appositamente dedicato alla prevedere la domanda futura ed ad eseguire calcoli delle quantità ideali da ordinare. I risultati delle trattazioni vengono resi disponibili mediante una dashboard interattiva, studiata per buyer e responsabili magazzino, che segnala in tempo reale le priorità operazioni attraverso indicatori cromatici e proposte quantitativi.

La validità della soluzione è stata testata su database reali di Igim Store, che riguardano vendite passate e quantità di stock, e ha evidenziato un netto miglioramento rispetto alle procedure manuali oggi utilizzate. I processi di classificazione hanno espresso elevate performance predittive, riducendo sensibilmente i casi di stock-out e di overstock, mentre il modello SARIMA ha garantito previsioni più precise rispetto ai processi alternativi con logiche fisse. In qualche modo, complessivamente, l'esecuzione ha permesso di incrementare l'efficienza operativa, contenere i costi di gestione delle scorte e aumentare il grado di livello di servizio erogato ai clienti. La ricerca dimostra quindi come un approccio data-driven, basato sull'integrazione di strumenti Business Intelligence e modelli predittivi, possa essere un vantaggio competitivo per e-commerce multicanale, aprendo orizzonti di ulteriore sviluppo in termini di automazione decisionale e scalabilità verso altri reparti aziendali.

Capitolo 1 - Introduzione	1
1.1 Contesto e obiettivi della ricerca	1
1.2 Struttura della tesi	3
Capitolo 2 - Inquadramento teorico	5
2.1 La gestione delle scorte nella supply chain e nell'e-commerce	5
2.2 Tecniche tradizionali di riordino e loro limiti	5
2.3 Business Intelligence e Data Analytics applicati alla logistica	8
2.4 Previsioni della domanda e modelli predittivi evoluti	8
Capitolo 3 - Analisi del caso IGLM.Store	11
3.1 Presentazione dell'azienda	11
3.2 Descrizione dei canali di vendita (eCommerce, Amazon FBM, Amazon FBA)	12
3.3 Gestione attuale delle scorte e del riordino	14
3.4 Criticità e problemi riscontrati	16
Capitolo 4 - Soluzione Proposta e Sviluppo degli Algoritmi	19
4.1 Architettura del Sistema di Business Intelligence	19
4.2 Fonti e Integrazione dei Dati	21
4.3 Parametri Strategici Considerati	25
4.3.1 Disponibilità attuale dei prodotti	25
4.3.2 Vendite storiche recenti	25
4.3.3 Presenza nella Buy Box	26
4.3.4 Prezzo BuyBox e prezzo interno	26
4.3.5 Ranking del prodotto su Amazon	27
4.4 Definizione dei Requisiti Funzionali	28
4.5 Algoritmo di Classificazione	30
4.5.1 Creazione del dataset di training	31
4.5.2 Selezione del modello	31
4.5.3 Preprocessing e bilanciamento	32
4.5.4 Addestramento e validazione	33
4.6 Algoritmo di Forecasting	34
4.6.1 Il modello SARIMA	35
4.6.2 Implementazione in Python	36
Capitolo 5 - Implementazione del sistema	37
5.1 Progettazione della dashboard di monitoraggio	37
5.2 Interfaccia per buyer e responsabili di magazzino	39
5.3 Logica di visualizzazione	42
5.4 Integrazione con i processi aziendali	46
Capitolo 6 - Validazione e risultati	51
6.1 Test su dataset storici e simulazioni	51

6.2 Analisi delle performance degli algoritmi	52
6.3 Risultati dei modelli di classificazione	53
6.3.1 Decision Tree	54
6.3.2 Random Forest	58
6.3.3 XGBoost	60
6.3.4 Confronto tra i modelli di classificazione	62
6.4 Risultati del modello di forecasting (SARIMA)	63
Capitolo 7 - Conclusioni	65
7.1 Risultati principali	65
7.2 Limiti dello studio	65
7.3 Sviluppi futuri	66
7.4 Contributo pratico per IGLM.Store	66
7.5 Considerazioni finali	67
Bibliografia	68

Capitolo 1 - Introduzione

1.1 Contesto e obiettivi della ricerca

Negli ultimi anni, l'e-commerce ha vissuto una crescita esponenziale, ridefinendo in profondità i modelli di acquisto dei consumatori e le logiche di distribuzione delle imprese. In un contesto caratterizzato da un'elevata competitività e da consumatori sempre più esigenti, la gestione della supply chain non può più essere considerata una funzione di supporto, ma rappresenta un elemento strategico per il successo aziendale, garantendo la disponibilità dei prodotti ed evitando gli sprechi ed i costi superflui.

La disponibilità costante dei prodotti e la rapidità delle consegne incidono direttamente sulla soddisfazione del cliente, sulla reputazione del venditore e sul posizionamento competitivo nei marketplace digitali. Tuttavia, la gestione delle scorte nell'e-commerce risulta particolarmente complessa a causa della forte variabilità della domanda, dell'influenza di fattori esterni come stagionalità, trend di mercato e promozioni, e della coesistenza di canali logistici differenti. Le metodologie tradizionali di riordino, basate su modelli statici e logiche deterministiche, non risultano più essere adeguate ad affrontare queste sfide, poiché incapaci di adattarsi dinamicamente ai cambiamenti del mercato e di sfruttare in modo integrato la mole crescente di dati a disposizione delle imprese.

In questo scenario si colloca il caso di IGLM.Store, e-commerce italiano con un vasto catalogo di articoli che spazia dalla fotografia ed elettronica fino a prodotti per la cura della persona, il fai-da-te e molto altro. L'azienda, con sede a Massa Lubrense (NA), funge da punto di riferimento per professionisti e appassionati in diversi settori merceologici, offrendo un'ampia gamma di marchi e novità hi-tech tramite il proprio store online.

Il contesto specifico di IGLM.Store presenta ulteriori complessità dovute alla presenza di canali di vendita molteplici. Infatti, l'azienda vende sia tramite il proprio sito e-commerce, sia attraverso la piattaforma Amazon in due modalità: FBM (Fulfilled by Merchant), ovvero gestendo internamente logistica e spedizioni per gli ordini ricevuti da Amazon, e FBA (Fulfilled by Amazon) (Toogood, 2024), dove invece Amazon stessa immagazzina e spedisce i prodotti per conto di IGLM.Store. Questa configurazione implica la gestione di due livelli di scorte: il magazzino centrale di IGLM (per evadere ordini diretti e Amazon FBM) e lo stock

dislocato nei centri logistici Amazon per FBA. Una distribuzione non ottimale delle giacenze tra questi magazzini può ridurre la disponibilità di prodotti su Amazon (limitando la visibilità nella Buy Box) o, viceversa, lasciare merce inutilizzata nel magazzino centrale, mentre potrebbe generare vendite se trasferita presso Amazon.

I primi tentativi di automatizzare il processo si sono concretizzati nello sviluppo di algoritmi a logica condizionale, capaci di classificare i prodotti da riordinare, ma privi di capacità predittiva ed incapaci di considerare variabili esogene.

Alla luce di queste sfide, l'obiettivo della presente ricerca è quello di sviluppare un sistema avanzato di Business Intelligence e Data Analytics capace di ottimizzare il processo di riordino e la gestione delle scorte per IGLM Store. In particolare, la tesi mira a progettare un modello che integri i dati interni (vendite storiche, giacenze, tempi di consegna) con informazioni di mercato esterne. Attraverso tecniche di analisi predittiva, il sistema dovrà fornire indicazioni puntuali su quando e quanto ordinare un prodotto e dove allocare le scorte (magazzino centrale vs. Amazon FBA) per minimizzare sia il rischio di stock-out sia l'eccesso di inventario.

In sintesi, gli obiettivi primari possono essere definiti come segue:

- Identificare proattivamente le esigenze di approvvigionamento o riallocazione: sviluppare un algoritmo di classificazione in grado di suggerire per ciascun articolo se è necessario avviare un riordino fornitore, effettuare un trasferimento di scorte verso i magazzini Amazon FBA, oppure non intraprendere alcuna azione (situazione di stock adeguato). Questo permetterà ai responsabili di focalizzarsi tempestivamente sui prodotti “critici”.
- Prevedere la domanda ottimizzando le quantità da riordinare: implementare un algoritmo di forecasting evoluto che, dinanzi ad una eventuale necessità di riordino, analizzi i dati storici di vendita e le variabili di mercato per stimare la domanda futura e calcolare la quantità ottimale da ordinare. Tale operazione ha come obiettivo quello di evitare decisioni arbitrarie o basate solo sull'intuito, riducendo così il rischio sia di sovra-approvigionamento (overstock), sia di riordini insufficienti.

Attraverso il conseguimento di tali obiettivi, la ricerca intende dimostrare come un approccio data-driven possa migliorare sensibilmente la gestione operativa di un e-commerce

multicanale (McAfee et al., 2012). In particolare, ci si attende un incremento dell'efficienza logistica, una diminuzione delle rotture di stock e dei costi legati a scorte in eccesso, nonché un miglioramento del livello di servizio percepito dai clienti, grazie a maggior disponibilità di prodotti e tempi di evasione più rapidi.

In ultima analisi, l'implementazione di un sistema di questo tipo rappresenta un'opportunità strategica per IGLM.Store per gestire il proprio catalogo prodotti in modo più intelligente e competitivo, massimizzando le vendite e la soddisfazione del cliente.

1.2 Struttura della tesi

La tesi è articolata in sette capitoli, organizzati in modo da seguire un percorso che, a partire dal quadro generale, giunge alla validazione della soluzione proposta e alle prospettive future. Nel primo capitolo viene introdotto il contesto della ricerca, con la definizione degli obiettivi, della metodologia adottata e della struttura complessiva del lavoro; mentre il secondo capitolo approfondisce il quadro teorico di riferimento, analizzando i modelli tradizionali di gestione delle scorte, i loro limiti nell'e-commerce e le opportunità derivanti dall'integrazione di strumenti di Business Intelligence e Machine Learning. Il terzo capitolo è dedicato all'analisi del caso IGLM.Store, descrivendone il modello di business, i canali di vendita utilizzati, le attuali modalità di gestione delle scorte e le principali criticità riscontrate. Il successivo capitolo presenta l'architettura della soluzione proposta, i requisiti funzionali e lo sviluppo degli algoritmi di classificazione e forecasting, evidenziandone logiche e modalità di applicazione. Successivamente viene illustrata l'implementazione del tool e della dashboard, con particolare focalizzazione sulle funzionalità operative e sul supporto decisionale offerto a buyer e responsabili di magazzino. Segue poi la validazione del sistema e l'analisi dei risultati, mettendo in luce i benefici concreti derivanti dall'introduzione della soluzione. Infine, dopo una sintesi delle principali evidenze emerse, si analizzano i limiti e le criticità del lavoro e se ne delineano le prospettive future di ricerca e sviluppo, sia per l'azienda sia per la letteratura scientifica.

Capitolo 2 - Inquadramento teorico

2.1 La gestione delle scorte nella supply chain e nell'e-commerce

La gestione delle scorte costituisce uno degli aspetti centrali della supply chain, in quanto incide in modo diretto sulla continuità operativa e sulla capacità di un'impresa di soddisfare la domanda dei clienti. Le scorte non sono soltanto una riserva di prodotti, ma un vero e proprio capitale immobilizzato: rappresentano, infatti, un investimento che deve essere bilanciato con i costi di mantenimento, che includono le spese di magazzinaggio, i rischi di obsolescenza e i vincoli finanziari.

In una supply chain tradizionale, la gestione delle scorte ha sempre perseguito il duplice obiettivo di evitare roture di stock e di minimizzare i costi logistici. Tuttavia, con la diffusione dell'e-commerce, il tema ha assunto una dimensione strategica.

I consumatori digitali non solo si aspettano tempi di consegna sempre più rapidi, ma tendono a penalizzare immediatamente i venditori che non garantiscono la disponibilità dei prodotti. A differenza della distribuzione fisica, dove l'assortimento può essere compensato da alternative immediate presenti all'interno del punto vendita, sulle piattaforme online la mancanza di un articolo porta spesso alla perdita definitiva della vendita, con un impatto diretto sulle performance economiche e sulla visibilità del venditore.

Questa dinamica spiega perché, come osservano Tuli, Mohammed e Sachani, «*l'ottimizzazione dei processi di inventory management costituisce un fattore chiave per garantire l'efficienza della supply chain e ridurre i rischi di stockout e overstock*» (Tuli et al., 2024). In altri termini, l'e-commerce non ammette inefficienze prolungate: la gestione delle scorte diventa non soltanto un processo interno, ma un elemento percepito dal cliente attraverso la disponibilità immediata e l'affidabilità del servizio.

2.2 Tecniche tradizionali di riordino e loro limiti

Le aziende hanno a disposizione da decenni una serie di metodologie “classiche” per la gestione delle scorte e l'impostazione delle politiche di riordino. Tra le tecniche tradizionali di riordino più diffuse si possono citare (Chopra & Meindl, 2016):

- Punto di riordino (Reorder Point) e scorta di sicurezza: consiste nel fissare un livello di giacenza (soglia) sotto il quale far scattare un nuovo ordine di approvvigionamento. Tale livello è calcolato in base al consumo medio durante il lead time di fornitura, aggiungendo eventualmente una scorta di sicurezza (safety stock) per far fronte a variazioni inattese della domanda o ritardi di consegna. Quando le scorte scendono sotto il punto di riordino, si emette un ordine di acquisto per una quantità prefissata.
- Rifornimento periodico (Periodic review): anziché monitorare continuamente le giacenze, si ispezionano i livelli di stock a intervalli di tempo fissi (es. settimanalmente o mensilmente) e si effettuano ordini per riportare le scorte a un livello predefinito (sistema “min-max” o “ordine a livello”).
- Just-In-Time e modelli “pull”: filosofie come il Just-In-Time mirano a ridurre al minimo le scorte, sincronizzando la produzione e l'approvvigionamento con la domanda effettiva. In tali approcci, il riordino avviene “tirato” dalla domanda reale (ad es. tramite sistemi a kanban), con l'obiettivo di eliminare sprechi di magazzino. Questo metodo, però, richiede una supply chain molto affidabile e tempi di approvvigionamento rapidi, risultando difficile da applicare integralmente in contesti di domanda fortemente variabile come l'e-commerce.

Queste tecniche tradizionali hanno certamente rappresentato a lungo lo zoccolo duro della gestione operativa delle scorte. Tuttavia, presentano una serie di limiti intrinseci, soprattutto se applicate in scenari moderni ad alta volatilità e complessità:

- Parametri statici e mancanza di adattività: i modelli classici (come punto di riordino fisso, EOQ calcolato su dati medi storici, ecc.) assumono spesso che la domanda e i tempi di consegna si mantengano stabili o con variazioni contenute. In pratica, nell'e-commerce la domanda può cambiare drasticamente in pochi giorni a causa di trend online o campagne promozionali, rendendo obsoleti i parametri statici. Un punto di riordino tarato su consumi medi rischia di innescare ordini troppo tardi se la domanda cresce improvvisamente o, al contrario, può far riordinare troppo presto/ troppo quantitativo in periodi di calo.
- Reattività invece di proattività: le logiche tradizionali tendono a essere reattive. Si attende che le scorte scendano sotto una soglia per emettere un ordine, oppure si rifornisca periodicamente sulla base di ciò che è stato consumato. Questo può portare a rincorrere gli eventi invece di anticiparli: ad esempio, un sistema a review periodica

mensile potrebbe accumulare backorder se durante il mese la domanda supera le attese, poiché non si interviene fino al prossimo ciclo di ordine.

- Decisioni isolate e mancanza di visione integrata: spesso le tecniche tradizionali considerano ciascun magazzino o canale in modo isolato. Nel caso di IGLM.Store, ad esempio, applicare indipendentemente un punto di riordino per il magazzino centrale e per lo stock in Amazon FBA potrebbe portare a squilibri (magari merce esaurita su Amazon ma abbondante a magazzino, senza meccanismi automatici di riallocazione). I modelli multi-echelon, che ottimizzano congiuntamente i livelli di scorta su più livelli della rete distributiva, non erano comunemente implementati nei sistemi tradizionali.
- Supremazia del dato storico interno: le metodologie classiche di forecasting e riordino si basano principalmente sui dati storici di vendita dell'azienda e sulle ipotesi di lead time noti. Non vengono tipicamente considerate informazioni esogene come i trend di mercato, i dati di sell-out dei concorrenti, l'interesse online (ricerche web, social sentiment), ecc. Questa visione parziale può risultare miope in un mondo dove i segnali esterni influenzano enormemente la domanda.
- Approssimazioni e ipotesi semplificative: modelli come l'EOQ assumono ad esempio domanda costante nel tempo, costo unitario invariabile, assenza di vincoli di capacità, ecc. Queste semplificazioni raramente rispecchiano la realtà completa. Nell'e-commerce possono esistere vincoli di budget, promozioni future pianificate, interazioni tra prodotti (effetti sostitutivi o complementari) che i modelli tradizionali non catturano.

In definitiva, pur restando concettualmente importanti, gli approcci tradizionali di inventory management mostrano evidenti segni di inadeguatezza di fronte alle sfide attuali. L'uso di punti di riordino statici e lotti fissi (EOQ) in sistemi tradizionali si traduce spesso in livelli di inventario lontani dall'ottimo e in opportunità di vendita perse. Allo stesso modo, i metodi convenzionali faticano a trovare il giusto equilibrio tra minimizzare i costi di mantenere scorte in eccesso e assicurare la disponibilità necessaria a soddisfare la domanda. In molti casi le aziende finiscono per oscillare tra periodi di stock-out e periodi di over-stock, sintomo di una governance poco agile.

Queste considerazioni motivano la ricerca di soluzioni più evolute. La sezione successiva esamina come gli strumenti di Business Intelligence e l'analisi avanzata dei dati possano

intervenire per superare alcuni dei suddetti limiti, offrendo una gestione delle scorte più informata, flessibile e integrata.

2.3 Business Intelligence e Data Analytics applicati alla logistica

Per affrontare la crescente complessità della gestione delle scorte, le imprese hanno progressivamente adottato strumenti digitali di Business Intelligence (BI) e Data Analytics. La BI permette di raccogliere dati provenienti da fonti eterogenee, integrandoli e trasformandoli in informazioni utili al processo decisionale. L'obiettivo è garantire al management una visione complessiva e in tempo reale delle scorte, delle vendite e delle performance logistiche.

Un vantaggio centrale della BI risiede nella capacità di superare la mera dimensione descrittiva: attraverso strumenti di predictive analytics, infatti, le imprese possono anticipare i fabbisogni, stimare le vendite future e pianificare gli ordini con maggiore precisione. Questo passaggio, come sottolineano Akter e Wamba, rappresenta uno sviluppo fondamentale perché consente di *“trasformare grandi quantità di dati in conoscenza utile per decisioni tempestive ed efficaci”* (Akter & Wamba, 2016, 175).

A ciò si aggiunge la dimensione della prescriptive analytics, che non si limita a prevedere scenari, ma fornisce raccomandazioni sulle azioni da intraprendere. In ambito logistico, questo significa non solo sapere quando la domanda di un prodotto aumenterà, ma anche ricevere suggerimenti su come bilanciare le scorte tra magazzini diversi, ottimizzare i riordini o pianificare le spedizioni.

L'impatto di tali strumenti è evidente nella trasformazione del ruolo dei buyer e dei responsabili di magazzino: non più semplici operatori che reagiscono agli eventi, ma decisori strategici supportati da strumenti digitali in grado di anticipare e simulare scenari alternativi.

2.4 Previsioni della domanda e modelli predittivi evoluti

L'introduzione del Machine Learning (ML) nella gestione delle scorte ha segnato una vera e propria rivoluzione concettuale. Gli algoritmi di ML, a differenza dei modelli tradizionali, non si limitano a calcolare medie o ad applicare formule statiche, ma apprendono dai dati, individuano pattern complessi e migliorano progressivamente la propria capacità predittiva attraverso l'iterazione.

Tra le tecniche più utilizzate vi sono i modelli di regressione avanzata, gli alberi decisionali, i Random Forest e le reti neurali. Questi strumenti permettono di considerare simultaneamente molteplici variabili: non solo dati storici di vendita, ma anche indicatori esogeni come stagionalità, eventi promozionali, prezzi della concorrenza e persino variabili macroeconomiche. In questo modo, la previsione della domanda diventa un processo dinamico, capace di adattarsi rapidamente a scenari in continuo mutamento.

Yang Liu, docente universitario, ha evidenziato come i modelli di ML applicati al forecasting riducono significativamente il rischio di stockout, migliorando la precisione della pianificazione degli ordini e aumentando la capacità di risposta delle imprese (Liu et al., 2025). L'integrazione di tali modelli nella supply chain contribuisce a superare i limiti dei sistemi basati su logiche condizionali o su previsioni puramente storiche, aprendo la strada a un approccio data-driven, proattivo e flessibile.

Naturalmente, l'adozione del ML non è priva di sfide, infatti, tra i principali ostacoli vi sono la necessità di disporre di dataset ampi e accurati, la complessità delle infrastrutture tecnologiche richieste e le difficoltà legate alla trasparenza dei modelli più sofisticati, come le reti neurali profonde. Tuttavia, le esperienze delle imprese leader del settore mostrano che i benefici derivanti dall'adozione di sistemi predittivi e prescrittivi superano ampiamente i costi iniziali e le difficoltà implementative.

Capitolo 3 - Analisi del caso IGLM.Store

3.1 Presentazione dell'azienda

Iglm Store è una realtà italiana nel settore dell'e-commerce, con sede a Massa Lubrense, in provincia di Napoli. Nata come rivenditore specializzato in articoli di fotografia e ottica, l'azienda ha progressivamente ampliato il proprio catalogo prodotti, fino a coprire un'ampia gamma di categorie merceologiche. Oggi Igelm si presenta come store online multi-settore, rivolto sia a clienti privati che professionali. Il sito è un punto di riferimento per appassionati ed operatori nei campi foto e video, elettronica, bellezza e cura della persona, prodotti per animali domestici, tempo libero, fai-da-te e altro ancora. Nel catalogo compaiono articoli quali fotocamere digitali, videocamere, obiettivi e accessori fotografici, dispositivi elettronici di consumo, piccoli elettrodomestici per la casa, attrezzature per giardino e piscine, utensili di ferramenta, prodotti per il fitness e molto altro. Questa diversificazione merceologica indica la volontà di iglm.store di competere con i grandi marketplace offrendo un assortimento ampio, mantenendo però l'identità di azienda italiana radicata sul territorio.

Un aspetto distintivo dell'azienda è l'attenzione al cliente e la costruzione di una reputazione di affidabilità. La presenza di oltre 500 recensioni su piattaforme indipendenti come Trustpilot, con un punteggio medio superiore a 4 su 5, testimonia un buon livello di soddisfazione della clientela. L'azienda sottolinea anche la garanzia sulle spedizioni e il supporto post-vendita come elementi chiave del proprio servizio. Operando esclusivamente online, iglm.store punta dunque su prezzi competitivi ed un'alta qualità del servizio con l'obiettivo di fidelizzare gli acquirenti in un mercato, quello dell'elettronica e accessori, molto affollato e competitivo.

Dal punto di vista organizzativo, iglm.store mantiene un'infrastruttura logistica snella. Il fulcro è il magazzino principale situato presso la sede di Massa Lubrense (NA), da cui partono sia gli ordini destinati ai clienti finali che acquistano sul sito e-commerce, sia le spedizioni per gli ordini gestiti come venditore su Amazon (modalità FBM). In aggiunta, l'azienda si avvale della logistica di Amazon (FBA) per ampliare la portata del proprio business su Amazon: ciò significa che una parte delle scorte di IGLM viene stoccata nei centri di distribuzione Amazon, in Italia e possibilmente in Europa, ed utilizzata per evadere gli ordini dei clienti Amazon con consegna Prime. Questo modello ibrido comune a molti seller Amazon consente di beneficiare della capillarità e velocità di Amazon stesso per i prodotti ad

alto turnover, mantenendo allo stesso tempo anche un canale di vendita diretto dal proprio sito.

A ragion veduta, Iglm.store compete nel panorama e-commerce multi-settore offrendo una vasta scelta di prodotti e presidia sia il canale diretto, cioè il proprio sito internet, sia il marketplace Amazon, sfruttando allo stesso tempo la modalità di logistica interna, per controllo e margini, e quella affidata ad Amazon, per ampliare la base clienti e garantire spedizioni rapide. Questa duplice natura dei canali di vendita ha implicazioni rilevanti, come vedremo, sulla gestione delle scorte e sulle decisioni di riordino, poiché richiede di gestire la disponibilità di prodotto in più luoghi e piattaforme.

3.2 Descrizione dei canali di vendita (eCommerce, Amazon FBM, Amazon FBA)

I canali di vendita di IGLM.Store possono essere suddivisi in tre categorie principali, ciascuna con proprie caratteristiche operative:

- E-commerce proprietario (web store): www.iglm.store è il sito ufficiale dell'azienda (I.G.L.M. Electronic srl), attraverso cui i clienti possono procedere alla selezione e all'acquisto autonomamente. In questo caso l'azienda gestisce l'intero ciclo di vendita: il cliente ordina sul sito, il magazzino di Massa Lubrense gestisce la richiesta e prepara la spedizione mentre l'azienda cura la consegna tramite corrieri. Questo canale, non essendo soggetto a pagamenti di commissioni a marketplace, offre margini più alti ed un rapporto diretto col cliente, ma richiede di attrarre traffico sul sito e garantire un livello di servizio comparabile ai grandi attori dell'e-commerce.
- Amazon – modalità FBM (Fulfilled by Merchant): Iglm ha un account venditore su Amazon e lista molti dei suoi prodotti sul marketplace. In modalità FBM, l'azienda riceve gli ordini tramite la piattaforma Amazon, ma la logistica (pick & pack e spedizione) rimane a suo carico. Dunque, un ordine Amazon FBM viene preparato nel magazzino di Massa Lubrense come fosse un ordine e-commerce proprio e spedito al cliente finale. Amazon incassa il pagamento dal cliente e trasferisce il ricavato al venditore (al netto di commissioni). Il vantaggio di FBM è che l'azienda non deve mantenere scorte presso Amazon e ha più controllo su imballaggi e spedizioni, tuttavia i prodotti FBM spesso non hanno il badge Prime e possono risultare meno visibili o appetibili rispetto a quelli gestiti direttamente da Amazon.

- Amazon – modalità FBA (Fulfilled by Amazon): in aggiunta o in alternativa al FBM, Igim utilizza il servizio FBA per alcuni articoli. In questo caso, l’azienda periodicamente trasferisce una quantità di prodotti dai propri magazzini ai magazzini Amazon. Da quel momento, quando un cliente acquista su Amazon uno di questi articoli, sarà Amazon stessa a occuparsi della preparazione e spedizione dal proprio centro logistico, offrendo servizio Prime, come la consegna rapida garantita e assistenza clienti Amazon. Il canale FBA consente di aumentare fortemente la visibilità su Amazon (i prodotti Prime hanno più chance di vincere la Buy Box) e di migliorare l’esperienza di consegna per il cliente, a scapito di costi logistici maggiori e della necessità di *anticipare* stock ad Amazon. Inoltre, su FBA la disponibilità immediata del prodotto è cruciale: se un articolo va out-of-stock nei magazzini Amazon, di fatto non è più acquistabile con Prime fino al successivo rifornimento.

Per Igim, gestire questi tre canali in parallelo significa dover segmentare le proprie scorte: una parte rimane a magazzino centrale per la copertura degli ordini e-commerce e Amazon FBM, un’altra parte viene dislocata nei magazzini Amazon (FBA). La proporzione di stock da allocare a FBA rispetto a quella da tenere in casa è una decisione strategica continua, influenzata dal volume delle vendite previste su Amazon, dai costi di stoccaggio Amazon che penalizzano overstock in FBA, e dall’esigenza di mantenere comunque prodotto per vendite dirette o emergenze. Un errore nella pianificazione può portare a situazioni spiacevoli, come avere un prodotto fermo in casa ma esaurito su Amazon perdendo così vendite potenziali su quel canale, oppure, al contrario, inviare un’eccessiva quantità di stock ad Amazon, tale da rimanere invenduto e generando così costi di deposito aggiuntivi.

Attualmente, l’azienda cerca di bilanciare i canali analizzando lo storico delle vendite di tutti i canali. Ad esempio, articoli che vendono molto su Amazon vengono progressivamente spostati verso FBA in quantità maggiori, mentre prodotti di nicchia o a bassa rotazione vengono tenuti principalmente per la vendita diretta e listati su Amazon solo in FBM, così da non incorrere in costi di logistica aggiuntivi. Questa gestione però è svolta in modo manuale ed empirico, come vedremo, e rappresenta una delle aree di possibile miglioramento tramite strumenti analitici.

3.3 Gestione attuale delle scorte e del riordino

Per comprendere le criticità e le opportunità di ottimizzazione, è importante descrivere come Igml gestisce oggi il processo di riordino e di controllo scorte, senza l'ausilio dei sistemi avanzati che la tesi propone di implementare. In base alle informazioni raccolte presso l'azienda, la situazione attuale è la seguente:

- Doppio magazzino, monitoraggio separato: l'azienda opera con due “magazzini virtuali” principali, uno corrispondente al deposito centrale a Massa Lubrense e uno rappresentato dallo stock allocato nei centri Amazon FBA. Per il magazzino centrale, esiste un gestionale interno che traccia in tempo reale le giacenze di ogni SKU, gli arrivi da fornitori e le uscite per spedizioni e trasferimenti. Parallelamente, Amazon fornisce via seller central i dati sulle giacenze FBA per SKU (incluso quanto è disponibile, quanto è riservato in ordini cliente non ancora spediti e quanto è in transito tra magazzini Amazon). L'azienda, quindi, monitora giornalmente entrambe le disponibilità: lo stesso monitoraggio avviene spesso tramite esportazione di report e gestione con fogli di calcolo.
- Processo di riordino manuale: nonostante la mole di dati a disposizione, attualmente il riordino avviene manualmente, basandosi in gran parte sull'esperienza dei responsabili acquisti e su analisi *ad-hoc*. Infatti, a cadenze regolari, ad esempio ogni settimana o ogni due, il responsabile verifica, per ciascun fornitore, quali prodotti del magazzino centrale siano sotto una determinata soglia e, sulla base di ciò, valuta l'emissione di un ordine di approvvigionamento. Tale soglia non viene calcolata con un algoritmo fisso ma deriva dall'osservazione dello storico delle vendite. Per Amazon FBA, in modo simile, si controllano gli SKU che stanno per esaurirsi nei fulfillment center e si decide se inviare un rifornimento da Massa Lubrense.
- Assenza di integrazione previsionale: attualmente non viene utilizzato alcun software di forecasting avanzato né vengono integrati dati di mercato esterni nel processo decisionale. Le uniche informazioni considerate sono lo storico delle vendite interne, spesso guardando ai registri degli ultimi 90 giorni e le giacenze attuali. Non si considerano, ad esempio, trend stagionali dell'anno precedente, né indicatori come il ranking Amazon o il numero di competitor attivi su un prodotto. Questo significa che eventuali variazioni imminenti della domanda potrebbero non essere anticipate.

- Gestione del trasferimento a FBA: anche la movimentazione delle scorte verso Amazon FBA avviene in modo reattivo. Infatti, nel momento in cui il team si accorge che un prodotto FBA è in esaurimento, prepara una spedizione di refill. Tale spedizione è tipicamente dimensionata sulla base della quantità disponibile a magazzino e del numero di vendite su Amazon nei mesi precedenti. Non esiste un sistema ottimizzato che suggerisca quali e quanti prodotti trasferire preventivamente a parte quello di proprietà di Amazon, che spesso non fornisce dati vantaggiosi per l'azienda. A ciò si aggiunga il possibile verificarsi di una vendita in quantità minori rispetto a quanto previsto, e quindi inviato; tali prodotti, qualora non vengano poi venduti, trascorsa una quantità di tempo e quindi, a fronte di ulteriori costi di stoccaggio, vengono fatti ritirare dall'azienda stessa.
- Indicatori di performance rilevati: i principali KPI, seguiti grossolanamente, costituiscono il tasso di rotazione delle scorte in magazzino centrale, il numero di stock-out verificatisi, conteggiando gli ordini annullati o le richieste in evase, ed il valore complessivo di inventario a fine mese. Non sono impostati KPI formali come livello di servizio per categoria, giorni di copertura scorte, ecc., sebbene a livello direzionale l'obiettivo sia minimizzare le rotture di stock, senza superare certi livelli di capitale allocato in magazzino.

Questa gestione artigianale ha il vantaggio di sfruttare la conoscenza tacita del personale (che conosce i prodotti di punta e i bestseller stagionali), ma presenta diversi punti deboli. In particolare, l'azienda riconosce che il processo attuale porta a problemi ricorrenti:

- Carenza di prodotti e stock-out: affidandosi a controlli periodici manuali, non sempre si riordina in tempo. Ci sono stati frequenti casi in cui un articolo è andato esaurito, sul sito e/o su Amazon, prima che venisse emesso il nuovo ordine al fornitore. Ciò ha causato perdite di vendite e peggioramento di posizionamento su Amazon: un prodotto out-of-stock perde la Buy Box e scende nei ranking di ricerca
- Eccessi di scorte e overstock: al contrario, per articoli a domanda incerta, si è talvolta riordinato in quantità superiori al necessario, ritrovandosi con scorte troppo elevate da smaltire. Questo aumenta i costi di mantenimento come spazio, capitale fermo, e nel caso di FBA, comporta anche costi diretti di storage presso Amazon, che dopo un certo periodo possono crescere in maniera significativa.

- Trasferimenti non ottimali tra magazzini: la gestione del trasferimento dei prodotti dal magazzino centrale a FBA non è sempre tempestiva né basata su criteri oggettivi. Si sono verificati ritardi nella disponibilità dei prodotti su Amazon FBA che ha ridotto la visibilità e le vendite su quel canale. Viceversa, sono presenti in FBA alcuni prodotti con giacenze eccedenti che generano costi evitabili.
- Carico di lavoro e possibilità di errore: il processo manuale richiede tempo e attenzione; con un catalogo che conta migliaia di SKU attivi, è facile perdere di vista qualche referenza. Non essendoci un sistema automatizzato di alert, l'efficacia dipende dalla diligenza individuale. Inoltre, l'assenza di integrazione significa dover controllare separatamente più sistemi, quindi il gestionale interno e la piattaforma Amazon, con rischio di errori di trascrizione o valutazione.
- Mancato sfruttamento dei dati di mercato: ignorando informazioni esterne come l'andamento del prezzo o il numero di concorrenti su Amazon, l'azienda può prendere decisioni subottimali. Ad esempio, se un competitor importante esce dal mercato di un certo prodotto, la domanda potenziale per IGLM potrebbe crescere e sarebbe il momento di aumentare lo stock, ma questo attualmente non verrebbe colto in tempo. Allo stesso modo, non cogliere un trend crescente di ricerca di un prodotto emergente significa perdere opportunità di essere pronti con le scorte.

3.4 Criticità e problemi riscontrati

Dall'analisi condotta sul caso, emerge un insieme di criticità nella gestione delle scorte e dei processi di riordino, molte delle quali riflettono limiti già evidenziati dalla letteratura teorica. Una prima difficoltà riguarda le frequenti rotture di stock: diversi articoli, in particolare quelli caratterizzati da un'elevata rotazione o da picchi improvvisi di domanda, vanno esauriti prima che un nuovo ordine venga processato e reso disponibile a magazzino. Ciò si traduce in perdite di vendita e, nel contesto Amazon, in una riduzione della visibilità del prodotto, poiché la sua indisponibilità indirizza i clienti verso alternative offerte dai concorrenti. L'assenza di un sistema di allert precoce accentua il problema, poiché il team si accorge della criticità solo quando l'articolo risulta già esaurito.

Un secondo elemento di criticità riguarda l'accumulo di scorte eccessive per prodotti a bassa rotazione. In questi casi, la paura di restare sprovvisti di merce, legata ai tempi di

approvvigionamento dai fornitori, spinge a mantenere livelli di stock troppo elevati. Ne derivano costi di magazzino non giustificati e giacenze che restano ferme per mesi, sia nel deposito centrale sia presso Amazon FBA, riducendo l'efficienza complessiva.

Vi è poi il problema dell'allocazione sub-ottimale tra magazzino centrale e logistica FBA: infatti, in mancanza di regole strutturate per decidere quando e quanti pezzi inviare ad Amazon, l'azienda sperimenta situazioni in cui lo stock risulta fermo a Massa Lubrense mentre i magazzini Amazon sono esauriti, con conseguente perdita di vendite, oppure al contrario accumula giacenze eccessive nei centri FBA, sostenendo costi di stoccaggio evitabili.

Il processo decisionale, inoltre, si fonda prevalentemente sull'esperienza e sull'intuito degli operatori, senza strumenti quantitativi evoluti di supporto. Ciò comporta inevitabili errori di valutazione, soprattutto considerando l'ampiezza del catalogo, che rende complesso mantenere una visione aggiornata e coerente su migliaia di referenze. Mancano simulazioni previsionali, indicatori precisi di copertura delle scorte o scenari alternativi di consumo, elementi oggi indispensabili in una gestione data-driven.

Tali criticità concorrono a delineare un modello gestionale prevalentemente reattivo, in cui si interviene solo quando il problema si manifesta. Non esistono meccanismi capaci di segnalare in anticipo situazioni a rischio, come l'esaurimento imminente di uno SKU o l'eccesso di giacenza di un prodotto a domanda debole. Di conseguenza, l'azienda è costretta a rincorrere continuamente le emergenze anziché anticiparle e governarle in maniera proattiva.

Le conseguenze delle criticità illustrate si riflettono poi su più fronti: perdita di ricavi per le vendite mancate, incremento dei costi legati a scorte ridondanti, urgenze logistiche e operazioni straordinarie per rimediare ai disallineamenti, oltre che ad un maggiore stress organizzativo per il personale. A ciò si aggiunga il rischio di compromettere la reputazione presso i clienti più fedeli, che potrebbero non trovare i prodotti desiderati, e di generare frizioni con i fornitori, spesso costretti a gestire ordini urgenti fuori pianificazione.

In conclusione, il quadro emerso dall'analisi conferma la presenza di ampi margini di miglioramento. Igim ha costruito i propri primi successi su un modello flessibile e sull'impegno diretto del team, ma per affrontare la crescita e gestire con efficienza un catalogo esteso su più canali necessita di strumenti e approcci più strutturati. È in questa

prospettiva che si colloca la proposta sviluppata nella tesi, orientata all'implementazione di un sistema di supporto decisionale basato su Business Intelligence e modelli predittivi avanzati. Tale soluzione mira a superare le criticità rilevate, trasformando l'approccio aziendale da reattivo a preventivo e garantendo una gestione più bilanciata delle scorte tra i diversi magazzini. Ci si aspetta che, grazie a questo sistema, l'azienda possa ridurre in maniera significativa i casi di stock-out e overstock, migliorare l'equilibrio delle allocazioni e costruire una supply chain più solida e in grado di sostenere lo sviluppo del business e-commerce.

Capitolo 4 - Soluzione Proposta e Sviluppo degli Algoritmi

4.1 Architettura del Sistema di Business Intelligence

Il sistema di Business Intelligence (BI) sviluppato è stato progettato con un'architettura modulare a più livelli, atta a integrare sorgenti di dati eterogenee e a fornire analisi avanzate per il supporto decisionale. In base ai principi generali dei sistemi BI, possiamo individuare alcune componenti fondamentali:

- le fonti dei dati, interne ed esterne
- i processi di estrazione, trasformazione e caricamento per l'integrazione dei dati in un deposito centralizzato
- un data warehouse o insieme di basi dati aziendali per la conservazione storica e l'organizzazione dei dati
- uno strato analitico contenente modelli e algoritmi (classificazione e forecasting)
- uno strato di presentazione che include la dashboard di monitoraggio delle scorte e dei suggerimenti operativi

Questa struttura rispecchia le best practice dei sistemi di BI, in cui le fonti dei dati primari, ad esempio il sistema ERP aziendale, e dei dati secondari, come quelli di mercato esterni, vengono individuate, raccolte e integrate. Successivamente, tali dati vengono puliti e memorizzati in un data warehouse centralizzato a supporto di analisi ed infine, metodi analitici come OLAP, data mining e forecasting, forniscono conoscenza utile ai decision-maker. L'architettura informatica realizzata garantisce quindi che i dati siano accessibili, affidabili e correttamente organizzati per le fasi successive di analisi e visualizzazione.

Dal punto di vista implementativo, le fonti dati interne comprendono il gestionale aziendale ERP, ed i database operazionali del magazzino principale bindCommerce, da cui si ottengono informazioni su scorte, ordini e vendite interne. Tali dati interni vengono estratti tramite connessione FTP automatizzata: il gestionale produce periodicamente file di esportazione in formato CSV contenenti, ad esempio, le giacenze di magazzino e lo storico delle vendite, che vengono depositati su un server FTP aziendale. Un processo schedulato provvede a scaricare questi file e ad importarli nel sistema BI. Parallelamente, le fonti dati esterne includono i dati provenienti da Amazon Marketplace e indicatori di mercato pertinenti. L'accesso ai dati di Amazon avviene attraverso le API ufficiali fornite da Amazon stesso per i partner di vendita,

Amazon Selling Partner API (Amazon Seller Central, 2025). Questo approccio garantisce affidabilità e conformità, in quanto l'Amazon SP-API rappresenta la modalità ufficiale per recuperare dati dagli account dei venditori Amazon, offrendo metodi documentati per accedere a informazioni su prodotti, ordini, inventario, prezzi e altre metriche. In particolare, attraverso l'API FBA Inventory e altre interfacce REST fornite da Amazon, il sistema recupera in tempo quasi reale la disponibilità delle scorte presso i centri logistici Amazon (FBA) e dati come il prezzo corrente o lo stato della Buy Box dei prodotti in vendita. È importante notare che l'utilizzo dell'API ufficiale garantisce un flusso di dati stabile e conforme alle policy Amazon, evitando pratiche di web scraping o l'uso di servizi esterni non autorizzati. I dati provenienti da Amazon includono quindi le quantità di stock disponibili nei magazzini di Amazon FBA, eventuali quantitativi in transito o riservati nei centri logistici, per tener conto di trasferimenti interni ad Amazon, nonché informazioni di contesto commerciale come il prezzo di vendita, la presenza nella Buy Box ed il ranking di prodotto.

Le informazioni estratte da fonti interne ed esterne vengono fatte confluire in un database di integrazione all'interno del sistema BI. In fase di ETL (Extract, Transform and Load), i dati vengono puliti, normalizzati e arricchiti. Ad esempio, eventuali discrepanze nei codici SKU(Stock Keeping Unit) tra ERP e Amazon sono risolte tramite tabelle di mapping in modo da poter unire correttamente i dataset: nel caso in esame, gli SKU interni corrispondono allo SKU utilizzato nel sistema ERP interno. Gli strumenti ETL implementati sotto forma di script Python pianificati, eseguono le operazioni di unione e pulizia ed i campi non utilizzati o incompleti vengono rimossi o imputati: nel dataset iniziale, ad esempio, la colonna "PREZZO NS." relativa al prezzo di listino interno presentava molti valori nulli ed è stata quindi eliminata in fase di pre-processing. Sulla base della stessa logica di coerenza necessaria, i formati dei dati, come date, codici categoria, ecc., sono stati resi in maniera omogenea. Questo processo garantisce che il data warehouse contenga dati storici accurati, consistenti e strutturati, condizione essenziale per qualunque sistema di BI.

Una volta che i dati sono caricati e aggiornati nel deposito centralizzato, il sistema passa alla fase analitica. Il cuore della soluzione consiste in due algoritmi principali: un algoritmo di classificazione e un algoritmo di previsione. Tali algoritmi sono implementati in Python e operano sui dati consolidati del warehouse. L'architettura prevede che i risultati prodotti dagli algoritmi ossia le etichette o raccomandazioni per ciascun prodotto e le eventuali quantità ottimali da riordinare stimate vengano memorizzate in apposite tabelle di output. Queste

tabelle alimentano a loro volta la dashboard di Business Intelligence, sviluppata come interfaccia per gli utenti finali, buyer e responsabili di magazzino.

La dashboard di monitoraggio rappresenta l'ultimo livello architettonale, presentation layer, e fornisce una vista riassuntiva e schematica dello stato delle scorte e delle raccomandazioni generate dal sistema. Nella dashboard, ogni prodotto preso in esame viene mostrato con una sintesi di informazioni chiave ossia SKU, descrizione, giacenze attuali in ciascun magazzino, vendite recenti, ecc., e soprattutto con l'indicazione dell'azione suggerita dal sistema: i prodotti per cui è necessario un riordino dal fornitore saranno contrassegnati dal colore rosso, quelli per cui è consigliato un trasferimento di scorte dal magazzino centrale a FBA in blu, mentre i prodotti con scorte adeguate, per i quali quindi non è necessaria alcuna azione, appariranno evidenziati in verde. Questo schema cromatico, definito in fase di progettazione dei requisiti, consente agli utenti di cogliere immediatamente le priorità. Ad esempio, il buyer aziendale visualizzerà subito l'elenco degli articoli critici in rosso da riassortire, mentre il responsabile del magazzino vedrà evidenziati in blu gli articoli da spedire con priorità ad Amazon FBA. L'architettura del sistema, riassumendo, combina dati interni ed esterni attraverso un robusto processo di integrazione con automatismi su FTP e API ufficiali, li centralizza in una base dati storica, applica algoritmi di machine learning e di time series forecasting per generare conoscenza ed infine la rende fruibile mediante un'interfaccia interattiva. Questo approccio riflette un tipico framework di BI ben strutturato (Evans, 2023), in cui ciascun componente svolge un ruolo chiave: le sorgenti dati forniscono la materia prima, ossia i dati grezzi, gli strumenti di integrazione ETL combinano e puliscono l'informazione garantendo qualità e pulizia dei dati, il data warehouse funge da serbatoio storico organizzato e gli analytics trasformano i dati in informazioni utilizzabili. L'utilizzo congiunto di fonti interne, come le giacenze ERP, ed esterne, ad esempio i dati forniti da Amazon Marketplace, offre una visione completa del contesto operativo e di mercato.

Come osservato in letteratura, integrare dati esterni nel processo di forecasting consente di migliorare accuratezza e reattività delle previsioni, riducendo rotture di stock e sprechi. Nel nostro sistema, ciò si concretizza nell'inclusione di parametri di mercato, come le vendite su Amazon, ranking, competitor, ecc., nel modello decisionale.

4.2 Fonti e Integrazione dei Dati

Il sistema sfrutta due principali fonti di dati: da un lato i dati interni dell'azienda, gestionale e magazzini, e dall'altro i dati esterni provenienti dal marketplace Amazon e correlati. La corretta integrazione di tali fonti è stata cruciale per ottenere una visione unificata e coerente dello stato delle scorte e della domanda.

Per quanto riguarda i dati interni, la base informativa interna comprende tutte le informazioni sulle scorte e sulle vendite aziendali storiche. In particolare, il focus è sul deposito di Massa Lubrense, che costituisce il magazzino centrale e che gestisce gli ordini dell'eCommerce proprietario e le vendite Amazon in modalità FBM, Fulfilled by Merchant, vendite spedite dal venditore. Dal gestionale ERP sono stati estratti storici di vendita, in particolare le quantità vendute per prodotto su base mensile negli ultimi 12 mesi, e le disponibilità di magazzino attuali. Inoltre, grazie alle Amazon API si rilevano le quantità in transito verso Amazon FBA, cioè spedite ma non ancora disponibili. L'estrazione avviene tramite connessione FTP automatizzata: il sistema ERP esporta giornalmente durante la notte un file CSV contenente per ciascun SKU le giacenze attuali nel magazzino centrale, mentre altri file contengono lo storico vendite. Questi file vengono scaricati dallo script pianificato e importati nel database BI. In fase di integrazione è stato svolto un importante lavoro di allineamento degli identificativi SKU, elemento chiave per unire correttamente i dati interni con quelli esterni. Lo store, oltre a vendere prodotti singoli, propone anche bundle e combo, che sono stati gestiti tramite la creazione di liste derivate, facilmente aggiornabili dagli operatori. Questo approccio ha garantito una gestione flessibile e coerente delle varianti complesse, assicurando che gli SKU dei prodotti, inclusi i bundle, coincidessero con quelli utilizzati nelle API di Amazon, merchant SKU. Grazie a questa configurazione e al monitoraggio, si è verificato che per tutti i prodotti l'identificativo risultasse univoco e coerente tra i sistemi, evitando così problemi di join mancati.

Per la gestione dei dati esterni, quindi provenienti dalla piattaforma Amazon, la soluzione proposta prevede la raccolta di una serie di dati provenienti da Amazon sfruttando l'API ufficiale Amazon Selling Partner (SP-API). In particolare, sono state utilizzate due categorie di endpoint API: un API di inventario FBA per ottenere la quantità di stock disponibile presso i centri logistici Amazon, inclusi eventuali quantitativi in arrivo o riservati; e un API di vendite e ordini per ottenere informazioni sul venduto tramite Amazon e sull'andamento dei prodotti, di cui in parte sopperite dai dati interni di vendita, dato che per vendite FBM l'ERP registra l'ordine. Inoltre, per contestualizzare il mercato, si sono considerati dati come il

prezzo corrente del prodotto su Amazon e la presenza nella Buy Box. Attraverso l'API Amazon Product Pricing e le API per le offerte si può recuperare il prezzo attuale e verificare se IgIm Store detiene la Buy Box ovvero se rappresenta il venditore prevalente per l'acquisto immediato. Quest'ultimo dato è importante poiché, come noto, oltre l'80% delle vendite su Amazon avviene tramite il riquadro Buy Box (Chen et al., 2016); di conseguenza, se il venditore perde la Buy Box, è prevedibile un calo significativo delle vendite di quel prodotto. Tra i dati esterni figurano anche parametri ottenuti tramite strumenti di analisi Amazon, ad esempio ranking di categoria del prodotto, numero di concorrenti sullo stesso prodotto, volume di ricerca delle parole chiave principali. Nel progetto iniziale era prevista l'integrazione di metriche di mercato tramite Jungle Scout, un servizio terzo di market intelligence; tuttavia, nella soluzione finale si è preferito utilizzare dati disponibili tramite API ufficiali e dataset interni, per coerenza e affidabilità. Tutti questi dati esterni vengono recuperati con cadenza regolare, tipicamente giornaliera, mediante chiamate programmate alle API.

La costruzione del dataset di training unificato ha comportato la consolidazione di dati eterogenei in un unico insieme coerente, con particolare attenzione all'integrazione e alla qualità dei dati. Nel dataset finale, ogni riga rappresenta un singolo SKU, ovvero un codice univoco di prodotto, e raccoglie tutte le informazioni rilevanti su quell'articolo. I campi includono le vendite mensili storiche da giugno 2024 a luglio 2025, memorizzate come colonne separate etichettate con anno e mese, in formato YYYY_MM, così da fornire la serie temporale dell'andamento delle quantità vendute. Sono inclusi inoltre i prezzi di riferimento, ossia il prezzo Amazon BuyBox corrente e il prezzo interno del venditore, che congiuntamente offrono indicazioni sul contesto competitivo e sulla strategia di pricing. Sul fronte logistico, il dataset integra le giacenze di magazzino sia per la gestione diretta da parte del venditore (FBM) sia per la logistica Amazon (FBA), rappresentando la disponibilità di stock in entrambi i canali. I dati categoriali associati a ciascuno SKU comprendono la principale categoria Amazon di appartenenza del prodotto (CAT1) ed eventualmente una categoria secondaria (CAT2), accompagnate dai rispettivi ranking di vendita nelle relative classifiche (RANK1 e RANK2) che indicano la posizione del prodotto in termini di performance commerciale all'interno di tali segmenti di mercato. È presente inoltre un indicatore della presenza o meno dell'offerta del venditore nella BuyBox di Amazon, un fattore critico che influenza la visibilità del prodotto e il volume di vendite. Infine, a completamento del dataset, è presente una colonna di etichetta predittiva denominata

“ALERT_30”, la quale sintetizza l’azione consigliata per lo SKU in un orizzonte di 30 giorni ai fini della gestione delle scorte e delle vendite. Tale campo può assumere valori come “Riordinare”, suggerendo di rifornire lo stock, “Trasferire”, indicando la necessità di ricollocare scorte tra magazzini o canali, oppure “OK”, dove nessuna azione è necessaria, e costituisce la variabile target per l’addestramento del modello predittivo. Nel processo di integrazione è stata posta grande cura anche alla qualità dei dati: sono state eliminate eventuali colonne interamente nulle o prive di valore informativo, ad esempio il campo di prezzo interno “PREZZO NS.”, qualora fosse risultato completamente vuoto, e sono state rimosse le righe prive dei dati essenziali per l’analisi, ad esempio SKU senza alcun dato di vendita o di giacenza. Inoltre, per gestire le informazioni mancanti senza introdurre distorsioni, si è proceduto all’imputazione dei valori nulli residui impiegando strategie appropriate, come la sostituzione con la media del rispettivo campo o con valori stimati in base al contesto. Tali accorgimenti di pulizia hanno permesso di ottenere un dataset finale coerente, affidabile e pronto per l’addestramento del modello.

Da un punto di vista logico, l’integrazione dei dati ha richiesto anche considerazioni sul *timing*: i dati interni di scorta e vendite sono aggiornati continuamente, mentre alcuni dati esterni, come il ranking Amazon, possono fluttuare quotidianamente. Si è quindi deciso di sincronizzare il tutto su base giornaliera, generando un aggiornamento completo ogni notte. In questo modo, ogni giorno la dashboard rispecchia le ultime informazioni disponibili: vendite fino al giorno precedente, scorte aggiornate e parametri di mercato aggiornati. Un’architettura a batch giornaliero è parsa sufficiente dato il contesto: infatti, le decisioni di riordino e trasferimento hanno orizzonte tipicamente settimanale o mensile, ragion per cui non richiedono un aggiornamento in tempo reale intraday. Qualora in futuro si presentasse l’esigenza di un sistema più real-time, l’architettura è predisposta per ricevere flussi dati in streaming, ad esempio tramite webhook dall’ERP o dall’API Amazon per variazioni di stock, e aggiornare la dashboard in modo ancor più tempestivo.

In sintesi, si è illustrano come si è scelto di rendere le differenti fonti informative interoperabili: il gestionale interno via FTP e i servizi Amazon via API rappresentano due mondi inizialmente separati, ma attraverso un’attenta progettazione dell’ETL è possibile ottenere un unico dataset integrato su cui costruire gli algoritmi. Tale dataset funge da base per tutte le analisi successive e contiene sia indicatori interni, affidabili indicatori della

performance storica dell’azienda, sia indicatori esterni che estendono la visibilità oltre l’azienda, includendo dinamiche di mercato.

4.3 Parametri Strategici Considerati

Il successo del sistema proposto dipende in larga misura dall’individuazione e dall’utilizzo appropriato delle *feature* che determinano le decisioni di riordino o trasferimento. Nell’ambito di ottimizzazione delle scorte per un’attività e-commerce multicanale, si è scelto di considerare un insieme di indicatori che combinino la dimensione interna con la dimensione esterna di mercato, come la competitività su Amazon e la posizione del prodotto nel marketplace. Di seguito si passano in rassegna i parametri principali inclusi nel modello, motivando l’importanza.

4.3.1 Disponibilità attuale dei prodotti

Rappresenta il numero di unità disponibili in ciascun magazzino rilevante. Nel caso specifico, distinguiamo Stock FBM e Stock FBA. Questi due valori sono fondamentali poiché definiscono lo stato attuale delle scorte. Un livello di stock basso o nullo nel magazzino, che serve un determinato canale, è il primo segnale di potenziale rottura di stock e quindi di necessità di riordino o trasferimento. Ad esempio, se Stock FBA è zero ma vi sono unità disponibili in FBM, potrebbe convenire trasferirne alcune ad Amazon prima possibile, tenendo conto dei tempi logistici, per evitare di perdere vendite su Amazon. Viceversa, se lo stock complessivo (FBM+FBA) è esiguo, sarà necessario un riordino dal fornitore. La soglia di esiguo viene valutata in relazione al consumo previsto (vedi vendite previste). Nel dataset finale questi valori compaiono come colonne FBM e FBA, e sono input diretti per l’algoritmo di classificazione.

4.3.2 Vendite storiche recenti

Nel dataset di training, le vendite sono registrate con cadenza mensile nel periodo compreso tra giugno 2024 e luglio 2025, con ciascun mese rappresentato da una colonna distinta ed etichettata secondo la convenzione anno_mese, in formato YYYY_MM. Questa struttura consente di ricostruire la serie temporale completa dell’andamento delle quantità vendute per ciascuno SKU, fornendo una base solida per l’analisi predittiva. A partire da tali dati, la dashboard operativa aggrega dinamicamente i valori su intervalli di 30, 60 e 90 giorni, così da offrire all’operatore una visione sintetica e immediatamente fruibile della velocità di rotazione

del prodotto. Lo stesso lavoro di aggregazione è stato effettuato anche per le vendite registrate nei corrispondenti 90 giorni dell'anno precedente, in modo da evidenziare eventuali pattern stagionali o ricorrenze cicliche. Questo indicatore permette di stimare la rapidità con cui un articolo si muove sul mercato e, in combinazione con le giacenze disponibili, di valutare per quanti giorni lo stock attuale potrebbe coprire la domanda. Sebbene nel modello implementato non sia stata applicata in maniera esplicita tale formula, l'algoritmo di classificazione è in grado di apprendere implicitamente la relazione tra i volumi di vendita recenti, insieme ai dati storici comparativi, e la necessità di riordino. Le vendite storiche a breve termine si configurano pertanto come un predittore robusto della domanda futura, a condizione che non intervengano variazioni stagionali improvvise o fattori esogeni straordinari.

4.3.3 Presenza nella Buy Box

La Buy Box di Amazon è il riquadro attraverso cui passa la stragrande maggioranza degli ordini Amazon. Il nostro parametro booleano indica se Igim Store detiene attualmente la Buy Box per quel prodotto, oppure la percentuale di tempo nelle ultime settimane in cui l'ha detenuta. Se l'azienda non è in Buy Box, cioè c'è un altro seller a prevalere, allora anche avendo stock disponibile su Amazon, è possibile che le vendite siano scarse perché i clienti acquisteranno dal concorrente. In tale scenario, il modello di classificazione potrebbe decidere che non conviene trasferire ulteriori scorte a FBA. Al contrario, se l'azienda è in Buy Box ed ha basso stock, allora tale condizione sarà da risolvere con priorità, tramite il riordino o il trasferimento, in quanto vi sarà opportunità di vendita immediata. Quindi la variabile BuyBox serve da moltiplicatore sul rischio di stockout: uno stock out, quando si ha la Buy Box, comporta perdita diretta di vendite, mentre uno stock out quando non si ha Buy Box potrebbe essere meno urgente (nel senso che le vendite erano già basse). L'inclusione di questo parametro rende il sistema più consapevole della competitività del prodotto sul marketplace.

4.3.4 Prezzo BuyBox e prezzo interno

Il dataset di training include la variabile relativa al prezzo BuyBox, ovvero il prezzo al quale lo SKU è proposto nella BuyBox di Amazon. Tale informazione è di primaria importanza, poiché la BuyBox costituisce il principale canale di visibilità e di conversione sulla piattaforma: la sua presenza, così come la sua assenza, incide in modo diretto sul volume delle vendite e, quindi, sulla necessità di mantenere adeguati livelli di scorte. L'analisi di questo prezzo permette di valutare la competitività dell'offerta rispetto al mercato e di comprenderne

le implicazioni sulla probabilità di stock-out o di accumulo di invenduto. Accanto a questo campo, nel dataset originario era presente anche la variabile *PREZZO NS*, relativa al prezzo interno del venditore, che tuttavia non è stata considerata nelle analisi a causa della mancanza sistematica di dati. La sua esclusione non ne riduce il potenziale valore informativo: in futuro, un'integrazione di dati completi su questo prezzo consentirebbe di arricchire il modello, introducendo un confronto più accurato tra la strategia di pricing interna e quella rilevata su Amazon, con ricadute positive sulla capacità predittiva sia degli algoritmi di classificazione sia dei modelli di forecasting.

4.3.5 Ranking del prodotto su Amazon

Amazon attribuisce ad ogni prodotto un Best Sellers Rank (BSR) in ciascuna categoria e sottocategoria di appartenenza, aggiornato frequentemente in base al volume di vendite recenti. Un rank basso, ad esempio 1, 50, 100, indica che il prodotto è tra i più venduti in quella categoria; un rank alto, ad esempio 100.000, indica vendite molto modeste. Nel dataset abbiamo due campi, RANK1 e RANK2, corrispondenti al rank nella sottocategoria specifica e nella macro-categoria principale. Questi valori offrono una misura della domanda relativa di quel prodotto sul mercato: se un articolo ha rank #1.102 nella macro-categoria “Giocchi e giocattoli”, significa che ci sono 1.101 prodotti più venduti in quella categoria, il che può comunque significare vendite discrete, mentre un rank #100.000 indicherebbe vendite minime. Il ranking è utile anche perché tende a riflettere rapidamente i trend: l'aumento di vendite migliora il ranking e viceversa. Nel nostro modello, un rank elevato potrebbe attenuare la necessità di riordino, mentre un rank molto basso è un forte segnale che serve tenere il prodotto sempre disponibile. Ricordiamo che il rank è una misura relativa all'interno di una categoria e viene considerato assieme ai dati di vendita assoluti per interpretarlo correttamente. Ad esempio, un rank #50 in “Illuminazione” implica un volume assoluto di vendite diverso da un rank #50 in “Libri”, data la diversa dimensione dei mercati, ma comunque serve come indicatore ordinalmente utile.

Riassumendo, i parametri strategici considerati nel modello integrato includono sia misure interne di performance come stock attuale e vendite storiche, sia misure esterne quali buybox, ranking, prezzo. Questa scelta è coerente con l'obiettivo di migliorare le decisioni di riordino combinando la conoscenza interna con la conoscenza esterna, incorporare indicatori di mercato esterni nel forecasting e nella pianificazione aiuta a cogliere segnali che i soli dati

interni non mostrerebbero. Nel nostro sistema, tutti i suddetti parametri vengono combinati nell'algoritmo di classificazione, come variabili esplicative nel modello di machine learning) per determinare l'azione consigliata. Inoltre, alcuni parametri, tra cui le vendite storiche, alimentano anche l'algoritmo di forecasting per calcolare la quantità ottimale da ordinare in caso di riordino necessario.

4.4 Definizione dei Requisiti Funzionali

All'avvio del progetto sono stati definiti precisi requisiti funzionali che la soluzione di Business Intelligence dovesse soddisfare, in linea con le criticità riscontrate nello scenario attuale (cfr. Proposta di Tesi) e con gli obiettivi prefissati. Tali requisiti hanno guidato sia la selezione dei dati e dei parametri visti sopra, sia lo sviluppo degli algoritmi, illustrati successivamente. Di seguito si passa ad elencare i principali requisiti funzionali, descrivendo come la soluzione implementata li realizza:

- RF1 – Integrazione di dati interni ed esterni: il sistema deve poter incrociare i dati interni di stock e vendite con informazioni esterne di mercato. Questo requisito nasce dalla necessità di avere una visione completa per prendere decisioni di riordino ottimali. La soluzione implementata soddisfa questo requisito attraverso il processo di ETL descritto precedentemente che aggrega le due fonti in un unico dataset. L'uso dell'API Amazon ufficiale e della connessione automatica al gestionale via FTP garantisce un flusso continuo di dati aggiornati, soddisfacendo il requisito di automazione nell'acquisizione dati. La sincronizzazione giornaliera assicura, inoltre, che le analisi siano effettuate su dati sempre aggiornati.
- RF2 – Monitoraggio delle scorte in tempo reale e rilevazione di sotto-scorta: il sistema deve monitorare continuamente i livelli di scorta nei magazzini e segnalare i prodotti sotto una soglia di sicurezza (sotto-scorta) per i quali è necessario agire. Questo requisito è perseguito tramite la dashboard e l'algoritmo di classificazione. In particolare, il modello di classificazione è addestrato proprio per riconoscere gli scenari di sotto-scorta (tenendo conto del tasso di vendita) e classificare quei prodotti come “Riordinare” o “Trasferire”. Il requisito di monitoraggio real-time viene soddisfatto dal colore assegnato ai prodotti in dashboard: un prodotto evidenziato in

rosso o blu equivale ad un alert di attenzione. L'utente può così immediatamente vedere quali SKU sono in condizione critica.

- RF3 – Raccomandazione sull'azione da intraprendere per ogni prodotto: ogni SKU deve essere corredata da un suggerimento automatico: *riordino* dal fornitore, *trasferimento* ad Amazon FBA, trasferimento ad Amazon FBA urgente, oppure *nessuna azione* se lo stock è sufficiente. Questo è il requisito centrale che ha portato allo sviluppo dell'algoritmo di classificazione multi-classe, in cui rientrano quattro classi: Riordino, Trasferimento, Trasferimento urgente, Ok. La classificazione è pensata per simulare la decisione che prenderebbe un responsabile logistica esperto: guardando vendite e scorte, decidere se riordinare, come nel caso di scorta globale insufficiente a coprire la domanda prevista, se semplicemente ridistribuire le scorte esistenti spostandone una parte su Amazon FBA, quando la scorta totale c'è, ma mal allocata, oppure se tutto è in equilibrio e non occorre fare nulla. La soglia tra queste decisioni è modulata su base statistica dal modello appreso. Il requisito è soddisfatto mostrando nella dashboard, accanto ad ogni prodotto, la label dell'azione suggerita in forma testuale e tramite il colore attribuito e precedentemente illustrato. Ad esempio: “*Riordinare*” per uno SKU esaurito sia in sede sia su Amazon, “*Trasferire*” per uno SKU con surplus in magazzino centrale ma out-of-stock su Amazon, “*Trasferire!!*” per uno SKU con out-of-stock più urgenti su Amazon, “*OK*” per SKU con livelli adeguati in entrambi i magazzini. Questo meccanismo aiuta i *buyer* a concentrare l'attenzione solo dove serve e a sapere esattamente cosa fare per ciascuno.
- RF4 – Calcolo della quantità ottimale da ordinare: nel caso in cui sia consigliato un riordino, il sistema deve fornire anche un quantitativo suggerito da acquistare dal fornitore, basato su analisi previsionali che minimizzino rischi di sovra-scorta o rotture. Questo requisito ha portato all'implementazione dell'algoritmo di forecasting. In pratica, per ogni prodotto marcato “Riordinare”, il sistema stima la domanda attesa per un certo orizzonte che si è ipotizzato di 12 mesi e, confrontandola con lo stock attuale, suggerisce quante unità ordinare per coprire quel fabbisogno. Ad esempio, se un prodotto vende in media 50 pezzi al mese e il lead time dal fornitore è di un mese, il sistema potrebbe suggerire di ordinare almeno 50 pezzi (oltre a un eventuale buffer di sicurezza). Il risultato viene mostrato anch'esso in dashboard in un *popup* specifico cliccando il tasto Forecast sulla corrispondente riga dello SKU. Il soddisfacimento di questo requisito si vede nel fatto che la soluzione non si limita a riordinare

semplicemente un prodotto “*riordina questo prodotto*”, ma analizza lo storico tramite i modelli ARIMA stagionali, per indicare “*riordina circa X pezzi*”, giustificando tale numero con l’andamento previsto.

- RF5 – Dashboard interattiva e facilità d’uso: il sistema deve presentare i risultati in modo chiaro, intuitivo e interattivo, permettendo all’utente di filtrare per priorità, cercare un prodotto e di visualizzarne i dettagli. Tale requisito ha guidato l’intera progettazione della dashboard, che così strutturata, consente l’ordinamento degli SKU secondo diversi criteri e fornisce, per ciascun SKU, tutte le informazioni necessarie per prendere una decisione corretta. Inoltre, direttamente dalla dashboard, è possibile impostare la scorta minima e consultare i forecast di ogni prodotto. Tali funzionalità assicurano l’accessibilità del sistema, cosicché anche un utente poco esperto può facilmente interpretare i colori e le indicazioni e prendere decisioni informate.

Oltre a questi requisiti principali, vi erano vincoli non funzionali come la sicurezza (gestione sicura delle credenziali API), la performance (il calcolo delle raccomandazioni deve avvenire in tempi rapidi, possibilmente pochi minuti per l’intero dataset), e la documentazione (tutti i risultati dovevano essere tracciabili e spiegabili in ottica di tesi, ad es. fornendo grafici e tavole di output). Tali aspetti sono stati curati secondo le seguenti modalità: le credenziali API sono memorizzate in modo sicuro e non esposte nel codice condiviso; il tempo di elaborazione per 2000 SKU è risultato nell’ordine di pochi secondi per la classificazione e pochi secondi per forecast per prodotto.

In conclusione, la soluzione progettata rispetta i requisiti funzionali prefissati, fornendo un sistema integrato che allerta l’utente sulle criticità di stock e lo guida nelle decisioni. Nei prossimi due paragrafi entreremo nei dettagli tecnici di come ciò avviene: prima illustrando l’algoritmo di classificazione e poi l’algoritmo di forecasting, evidenziando la teoria sottostante e le modalità di implementazione in Python.

4.5 Algoritmo di Classificazione

L’algoritmo di classificazione rappresenta il motore decisionale del sistema, incaricato di analizzare lo stato di ciascun prodotto e di assegnarlo a una delle quattro classi di azione: Riordinare, Trasferire, Trasferire!! oppure Nessuna azione (OK). In termini pratici, si tratta di un problema di classificazione multi-classe sulla base di variabili descrittive (features).

4.5.1 Creazione del dataset di training

Per affrontare un problema di classificazione supervisionata, il punto di partenza imprescindibile è la disponibilità di un dataset di training, ossia un insieme di osservazioni corredate da etichette che rappresentano la decisione corretta da intraprendere. In questo modo, l'algoritmo può analizzare la relazione tra le variabili descrittive dei prodotti e le scelte operative attese, sviluppando una regola generale in grado di essere applicata a nuovi casi. Poiché non si disponeva inizialmente di esempi etichettati, in quanto lo storico decisionale manuale dell'azienda non era formalizzato, si è proceduto a costruire un dataset di training etichettato applicando delle regole empiriche su dati storici, una sorta di etichettatura semi-supervisionata per poi addestrare modelli di machine learning supervisionato. In sostanza, si è simulato il comportamento che il responsabile avrebbe avuto e si sono create le classi: ad esempio, un criterio utilizzato è stato definire ALERT_30 = "Riordinare" se lo stock totale (FBM+FBA) era inferiore alle vendite previste dei prossimi 30 giorni (da qui la denominazione "ALERT_30"), ALERT_30 = "Trasferire" se lo stock Amazon era insufficiente ma esisteva stock in FBM a copertura (situazione di disallineamento), altrimenti ALERT_30 = "OK". Queste etichette iniziali hanno popolato la colonna ALERT_30 nel dataset, usata poi come variabile dipendente per il training. Il risultato è un insieme di circa 1900 prodotti con label, la distribuzione è risultata sbilanciata (molti prodotti erano in OK, pochi in situazioni critiche), quindi si tratta di un problema di classificazione sbilanciata a 4 classi.

4.5.2 Selezione del modello

Sono stati considerati diversi algoritmi di classificazione allo stato dell'arte per problemi multi-classe. Tra questi: Decision Tree, Random Forest, SVM, regressione logistica multiclass, k-Nearest Neighbors, e metodi ensemble come XGBoost. Data la natura dei dati e la necessità di interpretabilità ragionevole, si è poi preferito un approccio basato su alberi, Decision Tree e Random Forest. Gli Alberi Decisionali sono modelli predittivi che suddividono lo spazio dei dati in regioni omogenee rispetto all'etichetta, creando regole decisionali gerarchiche facilmente interpretabili. Un singolo albero però rischia di *overfittare* e può non cogliere bene tutti i pattern. Si è quindi valutato il modello Random Forest, un ensemble di alberi addestrati su sottocampioni casuali del dataset e sottoinsiemi random di feature, combinati tramite votazione per produrre la classificazione finale. Si è sperimentato anche XGBoost, Extreme Gradient Boosting, un metodo di boosting basato su alberi

decisionali che costruisce in maniera iterativa un ensemble di alberi riducendo l'errore residuo ad ogni iterazione. XGBoost è noto per le sue prestazioni elevate in molti compiti di classificazione e per la capacità di gestire efficacemente dataset di medie dimensioni con feature numeriche e categoriche tramite opportuna codifica. Nel nostro caso, XGBoost è stato provato come modello candidato, viste le dimensioni relativamente contenute del dataset e la necessità di curare lo sbilanciamento delle classi.

4.5.3 Preprocessing e bilanciamento

Prima dell'addestramento, il dataset è stato preparato attraverso alcune fasi. La fase iniziale del lavoro ha riguardato la pulizia del dataset, necessaria per garantire la qualità e l'affidabilità dei dati utilizzati nell'addestramento. In primo luogo, è stata rimossa la feature “PREZZO NS”, caratterizzata da un'elevata percentuale di valori mancanti che ne andavano a compromettere la significatività e l'utilità predittiva. Contestualmente, sono state eliminate alcune righe nulle che avrebbero introdotto rumore o inconsistenze nell'elaborazione successiva. Queste operazioni hanno permesso di ottenere una base dati più coerente e priva di elementi inutilizzabili, creando così le condizioni per le fasi successive di trasformazione e modellazione. Successivamente, è stato effettuato l'encoding delle variabili categoriali: ad esempio, le categorie di prodotto e l'etichetta ALERT_30 stessa, sono state convertite in valori numerici interi tramite *Label Encoding* (Géron, 2022). Questo assegna, ad esempio, 0 = OK, 1 = Riordinare, 2 = Trasferire, 3= Trasferire!!.. Un passaggio intermedio nella preparazione dei dati ha riguardato la verifica della presenza di outlier, ossia valori anomali che si discostano significativamente dal comportamento tipico delle variabili. La rilevazione è stata condotta sia mediante analisi descrittive, sia attraverso rappresentazioni grafiche come boxplot e distribuzioni di densità. L'obiettivo non era eliminare indiscriminatamente i valori elevati, che in alcuni casi riflettono picchi reali di domanda o particolari strategie commerciali, ma distinguere i casi plausibili da quelli derivanti da errori di registrazione o incongruenze nei dati. In seguito, per evitare che le variabili numeriche con scale diverse influenzassero in modo sproporzionato l'addestramento, si è proceduto alla loro standardizzazione. Sebbene i modelli basati su alberi in realtà non richiedono espressamente feature scaling, durante gli esperimenti è stato mantenuto per uniformità di pipeline, in particolare perché era prevista anche la possibilità di provare modelli come la regressione logistica o SVM che invece lo richiedono. Infine, si è affrontato il problema dello sbilanciamento delle classi: la classe “OK” costituiva la maggioranza dei casi. Per evitare che

il modello ignorasse le classi minoritarie, sono state tentate due strade: da un lato, nell’addestramento del Random Forest è stato usato il parametro di *class_weight*=“balanced” fornito da scikit-learn, che automaticamente pesa di più gli esempi di classe minoritaria (Riordinare, Trasferire) nell’*impurity* dei nodi. Dall’altro lato, si è sperimentata una tecnica di oversampling + undersampling combinati denominata SMOTEENN (Synthetic Minority Oversampling Technique + Edited Nearest Neighbors) (Batista et al., 2004). Questa tecnica genera sinteticamente nuovi esempi per le classi minoritarie (SMOTE) e contemporaneamente rimuove esempi ritenuti rumore dalle classi maggioritarie (ENN), al fine di bilanciare il training set in modo più netto. Nel codice è stata utilizzata l’implementazione di SMOTEENN dalla libreria *imbalanced-learn* (He & Ma, 2013). Dopo l’applicazione di SMOTEENN sul training set, la distribuzione delle classi è risultata pressoché uniforme, permettendo ai modelli di apprendere meglio anche le classi meno frequenti. Studi mostrano che SMOTEENN può migliorare le performance sui dataset sbilanciati, combinando i benefici dell’oversampling sintetico e della pulizia dei campioni rumorosi. Nel caso preso in esame, grazie all’utilizzo del dataset bilanciato, si è registrato l’aumento dell’accuratezza sulle classi “Riordinare” e “Trasferire”, a scapito di una leggera diminuzione sull’accuratezza della classe “OK”, che tutto sommato può ritenersi un accettabile compromesso essendo di maggior interesse non perdere segnalazioni di riordino/trasferimento, anche a costo di qualche falso allarme.

4.5.4 Addestramento e validazione

Il dataset è stato diviso in un set di training e uno di test (hold-out) usando l’80% per training e 20% per test, con stratificazione rispetto alla classe, così che il test set contenga proporzionalmente esempi di tutte le classi. Sul training set, come accennato, sono stati provati vari modelli: un Decision Tree con grid search di profondità, un Random Forest (20 alberi inizialmente), una Logistic Regression multiclass (one-vs-rest), un KNN e XGBoost (Géron, 2022). Si è utilizzata la tecnica della validazione incrociata (k-fold cross validation) per stimare le prestazioni medie e regolare gli iperparametri (Kohavi, 2001). Ad esempio, per il decision tree si è effettuata una grid search sul parametro *max_depth* da 2 a 5 con validazione incrociata 5-fold, individuando la profondità ottimale che potesse massimizzare un criterio (f1-weighted). Il Random Forest, grazie anche al bilanciamento interno, ha dato buoni risultati; si è testata l’importanza di aumentare il numero di alberi (*n_estimators*) e l’uso

o meno del dataset bilanciato SMOTEENN. XGBoost è stato addestrato sia sul dataset originale sia su quello bilanciato.

La metrica di valutazione principale considerata è stata l'accuratezza sul test set e il F1-score (Géron, 2022) per ciascuna classe (in particolare per le classi minoritarie). In ottica del nostro caso d'uso, avere buoni precision e recall sulle classi “Riordinare” e “Trasferire” è importante: ad esempio, *precision* per “Riordinare” indica quanti dei prodotti segnalati da riordinare effettivamente ne avevano bisogno, evitando falsi positivi, che porterebbero a ordini inutili, mentre *recall* per “Riordinare” indica quanti dei prodotti da riordinare secondo verità vengono effettivamente segnalati (evitando falsi negativi, cioè prodotti critici non segnalati). Dato che l'etichettatura “di verità” deriva da una regola euristica, l'accuratezza assoluta va interpretata con cautela; comunque la valutazione sul test serve a scegliere il modello più affine a replicare quelle decisioni.

4.6 Algoritmo di Forecasting

Quando l'algoritmo di classificazione segnala che per un determinato prodotto è necessario un riordino dal fornitore, sorge spontanea la domanda successiva: *in quale quantità ordinare?* Un ordine troppo scarso porterebbe a nuove rotture di stock a breve, mentre un ordine eccessivo causerebbe sovraccarico di magazzino e costi inutili. Per affrontare questo problema è stato implementato un modulo di forecasting basato sul modello SARIMA (*Seasonal Autoregressive Integrated Moving Average*), scelto per la sua capacità di tenere conto di eventuali pattern stagionali nelle serie storiche delle vendite. Lo sviluppo del modulo di forecasting rappresenta un aspetto centrale ma ancora in fase embrionale del progetto. L'obiettivo è fornire uno strumento capace di stimare la domanda futura dei prodotti e di suggerire la quantità ottimale da riordinare, riducendo sia il rischio di stock-out sia quello di sovraccarico di magazzino. Tuttavia, le potenzialità di questo approccio sono oggi limitate da due fattori principali: la disponibilità di serie storiche troppo brevi e la mancanza di variabili esogene sufficientemente integrate nel modello.

In questa fase iniziale, l'algoritmo è stato concepito più come prototipo dimostrativo che come strumento definitivo di supporto alle decisioni. Le previsioni generate devono quindi essere interpretate come indicazioni di massima, utili a mostrare la direzione di sviluppo e a validare la fattibilità tecnica dell'approccio, piuttosto che come output già pronti per guidare il

procurement in maniera autonoma. Ciò nonostante, il modello implementato, basato su un approccio SARIMA, consente di effettuare una prima esplorazione delle dinamiche temporali delle vendite e di evidenziare il valore aggiunto che un modulo predittivo può portare alla gestione delle scorte.

4.6.1 Il modello SARIMA

Il modello SARIMA è un'estensione del classico ARIMA (Zhao, 2024), arricchito da componenti stagionali. Esso viene indicato con i parametri $(p,d,q)(P,D,Q)$, dove (p,d,q) rappresenta rispettivamente l'ordine autoregressivo, il grado di integrazione (differenziazione) e l'ordine di media mobile per la componente non stagionale, mentre $(P,D,Q)_s$ rappresenta gli analoghi ordini per la componente stagionale con periodo s (ad esempio $s=12$ per dati mensili). In termini qualitativi, la parte AR del modello coglie la dipendenza dai valori passati, la parte I+, gestisce l'eventuale non stazionarietà della serie, rendendola stazionaria tramite differenziazione, e la parte MA modella gli shock o errori residui. L'aggiunta della componente stagionale consente al modello di riconoscere e prevedere cicli che si ripetono periodicamente (ad esempio, picchi di domanda in determinati mesi dell'anno).

Nel nostro caso, la serie storica analizzata per ciascun prodotto consiste nelle vendite mensili su un orizzonte temporale significativo per il business. L'obiettivo del modulo di forecasting è stimare quante unità di un prodotto dovranno essere riordinate nei periodi successivi, supportando così le decisioni di acquisto e il trasferimento delle scorte tra i magazzini.

Prima di procedere alla modellazione, ogni serie temporale è stata esaminata per identificare trend a lungo termine o stagionalità annuali. In particolare, sono state calcolate e analizzate le funzioni di autocorrelazione semplice e parziale, ACF e PACF, allo scopo di ottenere indicazioni sui possibili ordini p e q del modello ARIMA e sugli ordini stagionali P e Q da adottare. Parallelamente, è stato applicato il test statistico di Dickey-Fuller aumentato (ADF) (Bobbitt, 2021) per valutare la stazionarietà delle serie e determinare il grado di differenziazione d necessario: qualora una serie mostrasse un chiaro trend o ciclicità non stazionaria, si procedeva a differenziarla.

Stabiliti questi elementi, si è quindi proceduto ad adattare un modello SARIMA ad ogni serie storica di prodotto.

La calibrazione è avvenuta in modo iterativo: inizialmente sono stati impostati ordini di base dedotti dall'analisi dell'autocorrelazione, un modello SARIMA $(1,1,1)(1,1,1)_{\{12\}}$ è stato

scelto come punto di partenza per molte serie con stagionalità annuale, successivamente si è valutata la bontà dell’adattamento e si sono apportati aggiustamenti empirici ai parametri per migliorare le prestazioni predittive.

Una volta validato, il modello SARIMA addestrato per ciascun prodotto è in grado di generare previsioni sulle vendite, o sui fabbisogni di riordino, per un certo numero di mesi futuri prefissato, ad esempio i successivi 6 o 12 mesi. Tali proiezioni forniscono al sistema di supporto decisionale indicazioni quantitative su *quando* e *quanto* riordinare ciascun articolo, integrandosi con le altre logiche decisionali del sistema di inventory management descritte in precedenza.

4.6.2 Implementazione in Python

Il forecasting è stato implementato come appendice allo script di analisi, sfruttando *statsmodels.tsa.statespace.SARIMAX* (Seabold et al., 2010). Il codice segue i seguenti passi: estrae le prime 12 colonne di vendite dal dataframe, traspone per ottenere una serie temporale (righe come periodi), esegue il test ADF sulla prima serie, differenzia se necessario, mostra ACF/PACF (questi passi sono stati eseguiti per esplorazione ma non automatizzati per ogni SKU), quindi viene definito il modello SARIMAX con parametri scelti (nel codice: `order=(1,1,1)`, `seasonal_order=(1,1,1,12)` come base), esegue fit del modello e ottiene forecast per 12 mesi. Le previsioni negative vengono poste a zero, vengono poi plottati i dati storici e le previsioni sullo stesso grafico, come in Figura 4.3, ed infine i dati vengono esportati tramite un file .csv, che grazie al caricamento in cloud viene mostrato all’interno della dashboard.

Capitolo 5 - Implementazione del sistema

5.1 Progettazione della dashboard di monitoraggio

La figura 5.1 illustra una schermata esemplificativa della dashboard, in cui i prodotti sono colorati in base allo stato: si noti come i prodotti contrassegnati in rosso e blu emergano chiaramente, facilitando un colpo d'occhio sulle azioni da intraprendere.

SKU	NOME	EAN	BRITA	MASSA-CASERTA	DISP. FBA	IN ENTRATA-FBA	RISERVATA FBA	TOT. FBA	Vendite 30 Succ. Fbm + Etto	Vendite 30 Succ. Anno Prez. Pzo	Vendite 30 Succ. Anno Prez. Pzo	SCORTA MINIMA	ALERT
1038408	BRITA ON TAP NEW WATER SYSTEM 600L	4006387102159	BRITA	77	0	0	0	0	0	0	0	0	OK
1024050	BRITA KIT MARELLA CON 3 FILTRI MAXTRA+	4006387076771	BRITA	66	0	0	0	0	1	0	0	0	OK
1051435	BRITA ALUNA WHITE PRO ALL-IN-ONE		BRITA	42	11	0	0	11	0	0	53	8	Trasferire!!
● 1009007	BRITA ON TAP REFILL 1200L	4006387022938	BRITA	34	160	0	0	160	36	68	76	196	Riordinare
● 1023721	BRITA MARELLA WHITE 1 FILTRO	4006387076610	BRITA	23	0	0	0	0	8	0	0	72	Riordinare
1043355	Brita Marella White 2 filtro OMAGGIO Caraffe filtrante a	4006387095239	BRITA	10	0	0	0	0	0	0	0	0	OK
1041063	Brita Borraccia Filtrante Fill Go Lime 0.6 Litri	4006387061265	BRITA	4	128	0	0	128	0	0	129	169	Riordinare
BRITA COPERTO COPERCHIO DI													OK

Figura 5.1: Dashboard di monitoraggio con evidenziazione dello schema colori

In seguito allo sviluppo degli algoritmi descritti nei capitoli precedenti, è stata realizzata una dashboard di monitoraggio come interfaccia operativa del sistema. Si è scelto di utilizzare il framework web *Flask* di Python per implementare questa dashboard, poiché esso consente di creare prototipi rapidamente e di integrare agevolmente il motore analitico in un'applicazione web leggera. La dashboard rappresenta il livello di presentazione del sistema di Business Intelligence, rendendo accessibili agli utenti finali, i buyer e i responsabili di magazzino in Igim Store, i risultati delle elaborazioni in modo intuitivo e interattivo. In altri termini, dopo aver consolidato e processato i dati attraverso l'ETL e gli algoritmi di classificazione/forecasting, il sistema veicola le informazioni chiave agli stakeholder tramite un front-end web user-friendly.

La progettazione di questa dashboard è stata guidata dai requisiti funzionali individuati in fase iniziale. In particolare, si rendeva necessario uno strumento che presentasse in modo chiaro e sintetico lo stato delle scorte e le raccomandazioni operative generate dal sistema. Per soddisfare tale requisito, la struttura dell'interfaccia è stata studiata per offrire una vista

riassuntiva di ogni prodotto che deve essere monitorato, affiancando ai dati quantitativi anche indicatori visuali (colori, etichette testuali) delle azioni suggerite. Si è optato per una visualizzazione tabellare, ritenuta più adatta a mostrare molte informazioni per un insieme esteso di SKU in un'unica schermata. Ogni riga della tabella corrisponde a un prodotto del catalogo dell'azienda, mentre le colonne rappresentano differenti categorie di dati, identificativi, giacenze, vendite, indicatori di performance e suggerimenti decisionali. La Figura 5.1 illustra l'aspetto della dashboard: in essa si può notare come l'organizzazione tabellare permetta di confrontare rapidamente i prodotti e identificare a colpo d'occhio quelli critici, grazie all'uso combinato di ordinamento e codici colore.

Dal punto di vista tecnologico e architettonale, la dashboard Flask è stata implementata seguendo un approccio *Model-View-Controller* semplificato. La parte di modello è costituita dal database di Business Intelligence in cui risiedono i dati integrati e i risultati degli algoritmi, come le etichette di alert e le previsioni di domanda. A seguito dei processi di ETL notturni e dell'esecuzione degli algoritmi di classificazione e forecasting, le informazioni aggiornate vengono salvate in apposite tabelle di output. Queste ultime fungono da sorgente dati per la dashboard, assicurando che l'interfaccia mostri sempre lo stato più recente possibile. La componente di vista è rappresentata dalle pagine web generate da Flask: esse includono il codice necessario a visualizzare la tabella interattiva e gli elementi grafici. La logica di controllo, infine, è implementata nei route di Flask che gestiscono le richieste dell'utente e interagiscono con il modello per recuperare i dati da mostrare.

Un aspetto fondamentale in fase di progettazione è stata la definizione di uno schema cromatico efficace per segnalare le priorità. Si è deciso di utilizzare tre colori principali associati ai possibili stati di un prodotto: il colore verde indica uno stock adeguato per il quale non sono richieste azioni (situazione “OK”), il rosso, indica la necessità di riordino da fornitore, un’azione quindi urgente per evitare rotture di stock), mentre il colore blu indica la necessità di trasferimento di scorte dal magazzino centrale a Amazon FBA, un’azione mirata a riallocare stock lì dove serve.

Questo schema, definito durante la progettazione, permette agli utenti di cogliere immediatamente le urgenze semplicemente osservando i colori evidenziati. Ad esempio, un articolo evidenziato in rosso segnala al buyer un potenziale stock-out e dunque la priorità di emettere un nuovo ordine al fornitore; un articolo in blu avvisa il responsabile logistico che è opportuno trasferire quanto prima quel prodotto nei magazzini Amazon. I prodotti in verde

non richiedono attenzione immediata. Questa codifica a colori è risultata intuitiva anche per utenti non tecnici, in linea con il requisito funzionale numero 5, che sottolinea la facilità d'uso per utenti non esperti. Oltre ai colori, ogni riga mostra anche un'etichetta testuale esplicita (es. “Riordinare” o “Trasferire”) nella colonna *Alert*, così da evitare ambiguità e consentire anche una eventuale consultazione in bianco e nero, ad esempio stampe cartacee, senza perdere il significato delle segnalazioni.

In sintesi, la dashboard di monitoraggio è stata progettata su misura per le esigenze decisionali di IGLM.Store. La fase di design ha tenuto conto sia degli aspetti funzionali, quali dati mostrare, come organizzarli, come aggiornare le informazioni, sia degli aspetti ergonomici, facilità di lettura, immediatezza nella comprensione delle priorità, interattività. Il risultato è un’interfaccia che coniuga completezza informativa e semplicità d’uso: essa funge da punto di convergenza di tutta l’intelligenza analitica sviluppata, presentandola in modo da supportare attivamente i processi decisionali quotidiani.

5.2 Interfaccia per buyer e responsabili di magazzino

Fin dall’inizio, è stato chiaro che i principali utilizzatori della dashboard sarebbero stati due figure aziendali con esigenze complementari: il buyer, responsabile degli approvvigionamenti, e il responsabile di magazzino o logistica. La progettazione dell’interfaccia ha quindi considerato esplicitamente questi due ruoli, in modo che ciascuno potesse trarre rapidamente le informazioni di proprio interesse. È importante sottolineare che, pur utilizzando la stessa dashboard, buyer e responsabili di magazzino la fruiscono con prospettive diverse: il sistema è stato pensato per supportare entrambi in parallelo, evitando di dover creare due strumenti separati. Dal punto di vista del buyer, un aspetto rilevante dell’interfaccia è la possibilità, per ciascuno di esso, di configurare nel proprio profilo i brand autorizzati alla gestione. Questa scelta avviene direttamente all’interno della sezione dedicata al profilo utente, dove è possibile selezionare i marchi di competenza, come mostrato in figura 5.2.

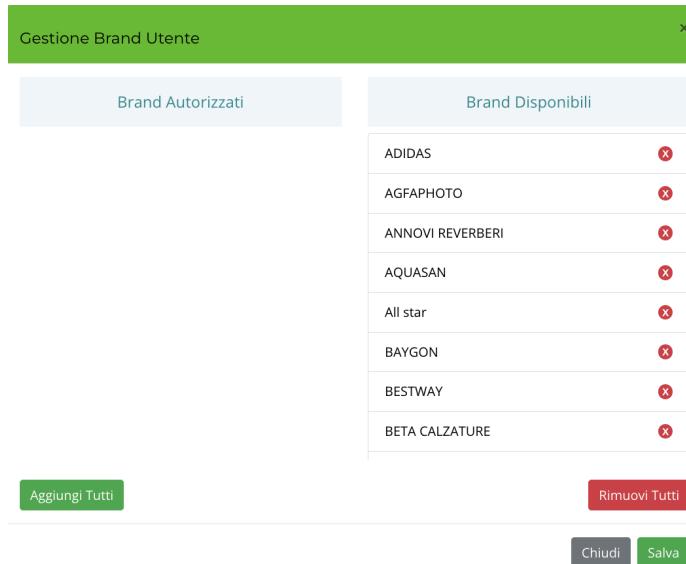


Figura 5.2: Popup scelta Brand Autorizzati

Tale configurazione diventa il riferimento operativo per il sistema: ogni volta che viene avviata una sessione di riordino, il tool genera un popup automatico che mostra l’elenco dei prodotti da riordinare limitatamente ai brand autorizzati per quello specifico buyer. Successivamente, una volta chiuso il popup, il buyer mantiene comunque la possibilità di visualizzare l’intero catalogo dei prodotti presenti nel sistema e di avviare il proprio lavoro di ricerca e analisi senza restrizioni. In questo modo, la funzionalità dei brand autorizzati funge da filtro iniziale e da supporto operativo, guidando il buyer verso le priorità relative ai marchi di sua competenza, ma senza limitare la visione complessiva del portafoglio prodotti. Ne risulta un’interfaccia flessibile, capace di coniugare personalizzazione e completezza, che permette di ottimizzare le attività di riordino e, al tempo stesso, di garantire un controllo integrale sulle scorte disponibili. Per ciascun prodotto, il buyer può consultare informazioni fondamentali per prendere decisioni informate: ad esempio le giacenze attuali nel magazzino centrale, le vendite recenti e la *scorta minima* impostata. In questo modo è possibile capire se un articolo sta per esaurirsi perché le vendite sono state elevate o perché le scorte iniziali erano basse. Il buyer può inoltre utilizzare la funzionalità di ricerca testuale integrata nella dashboard, per nome o SKU, per filtrare specifici prodotti o categorie merceologiche di suo interesse, e verificare se per essi sono richiesti riordini. Questa funzione è utile, ad esempio, quando il buyer vuole concentrarsi su una particolare linea di prodotti o su un fornitore specifico: digitando un termine di ricerca o filtrando per brand, la lista si riduce alle sole voci pertinenti, facilitando un’analisi mirata.

Dal punto di vista del responsabile di magazzino, la dashboard agevola l'individuazione degli articoli che richiedono un trasferimento a FBA. Questi appaiono evidenziati in blu, segnalando che, pur essendoci stock disponibile nel magazzino centrale, le scorte presso i centri logistici Amazon sono insufficienti per coprire la domanda su quel canale. Il responsabile di magazzino, visualizzando tali elementi, può prontamente pianificare una spedizione verso Amazon FBA per evitare possibili roture di stock sulla piattaforma Amazon. Anche in questo caso, tutte le informazioni rilevanti sono a portata di mano: la dashboard mostra infatti, sia la quantità disponibile nel magazzino centrale, sia quella attualmente disponibile e in transito presso Amazon FBA, distinguendo tra unità già disponibili, unità in arrivo a breve ed unità riservate per ordini in corso. In questo modo l'operatore può avere una visione completa anche sulla merce già inviata, ma non ancora attivata nel magazzino Amazon. Ciò consente di valutare non solo *se* trasferire, ma anche *quanto* trasferire, il responsabile può infatti confrontare le vendite mensili su Amazon con lo stock residuo e quello in arrivo, per stimare il fabbisogno di rifornimento di FBA. Come per il buyer, anche il responsabile magazzino può utilizzare la barra di ricerca o eventuali filtri per concentrarsi su determinati gruppi di prodotti. L'interattività della dashboard, dunque, offre ad entrambi i ruoli flessibilità nell'estrarrre le informazioni rilevanti per le rispettive responsabilità.

Un elemento chiave è che l'interfaccia promuove la collaborazione e la condivisione delle informazioni tra i due ruoli. Poiché buyer e responsabile di magazzino accedono alla medesima piattaforma, viene meno il rischio di disallineamenti informativi: tutti dispongono degli stessi dati aggiornati quotidianamente riguardo alle scorte e alle azioni raccomandate. In pratica, il buyer può vedere quali prodotti sono in trasferimento (blu) e comprenderne il motivo, ad esempio, se un articolo è contrassegnato “Trasferire”, saprà che in magazzino centrale c’è stock, ma Amazon sta esaurendo le scorte; analogamente il responsabile di magazzino può notare se un prodotto è anche da riordinare (rosso), capendo che dopo il trasferimento sarà comunque necessario un approvvigionamento esterno poiché la quantità globale è insufficiente. Questa visibilità incrociata aiuta a coordinare le azioni: il buyer, ad esempio, potrà dare precedenza a ordini fornitori per articoli che il magazzino sta trasferendo, così da ricostituire presto le scorte centrali, mentre il magazzino potrà prepararsi a trasferire parte degli articoli che il buyer sta per riordinare in grandi quantità, assicurando che una quota arrivi anche ad Amazon se la domanda su quel canale è forte.

Dal punto di vista della user experience, l’interfaccia è stata mantenuta il più semplice e uniforme possibile per entrambi gli utenti. Non è stato necessario sviluppare due interfacce separate, si è scelto invece un design unico, dove alcune funzionalità chiave, come l’ordinamento per priorità, i filtri ed i pulsanti di dettaglio, soddisfano le esigenze di entrambi i ruoli. L’uso dei colori permette di “personalizzare” la vista per ruolo in modo naturale: il buyer presterà maggiore attenzione alle evidenziazioni rosse, il responsabile magazzino a quelle blu, ma entrambi vedono l’intero quadro. Questa soluzione integrata favorisce anche un eventuale scambio di ruoli o di informazioni: se, per esempio, il buyer fosse assente, il responsabile magazzino potrebbe comunque identificare i prodotti da riordinare, poiché contrassegnati dal colore rosso e magari avvisare il team acquisti, e viceversa. In conclusione, l’interfaccia è stata concepita per essere multi-utente e multi-ruolo, garantendo a tutti gli attori coinvolti nella gestione delle scorte la possibilità di lavorare sugli stessi dati e con gli stessi strumenti decisionali, migliorando la coerenza e la tempestività delle azioni intraprese.

5.3 Logica di visualizzazione

La logica di visualizzazione adottata nella dashboard determina quali informazioni vengono mostrate agli utenti ed in che modo esse sono organizzate e messe in risalto. L’obiettivo primario è presentare una sintesi completa ma chiara dello stato di ciascun prodotto, in modo da supportare decisioni rapide e fondate. A tal fine, la dashboard combina dati di diversa natura (scorte, vendite, previsioni, KPI di mercato) seguendo un’organizzazione razionale e facilmente leggibile.

Innanzitutto, ogni riga della tabella presenta un identificativo univoco del prodotto e la sua descrizione: tipicamente vengono mostrati il codice SKU interno, il nome o descrizione commerciale dell’articolo e, ove disponibile, un identificatore universale come il codice EAN. Questa sezione iniziale permette all’utente di riconoscere immediatamente il prodotto di riferimento. Subito a fianco, la dashboard mostra le giacenze di magazzino suddivise per polo logistico:

- Stock magazzino centrale: una colonna dedicata riporta la quantità disponibile nel magazzino di Massa Lubrense sommata a quella presente nell’eventuale deposito di supporto. Nel prototipo tali due sedi sono aggregate in un unico valore “Massa+Caserta” per esprimere la disponibilità totale interna. Questo dato indica quante unità l’azienda ha fisicamente a disposizione per evadere ordini diretti e per

rifornire Amazon (modalità FBM).

- Stock Amazon FBA: accanto allo stock centrale, vengono mostrate diverse colonne relative alle scorte presso Amazon. In particolare, si distinguono almeno tre voci:
 - *Disponibile FBA*: unità attualmente disponibili nei magazzini Amazon (pronte per la vendita immediata su Amazon).
 - *In transito FBA*: unità che sono state spedite dalla sede centrale verso Amazon e risultano in consegna o in fase di ricevimento da parte di Amazon (non ancora disponibili alla vendita).
 - *Riservato FBA*: unità attualmente allocate ma non disponibili per la vendita perché impegnate (es. prodotti già venduti ma non ancora spediti, oppure unità bloccate in trasferimenti interni tra centri Amazon).
 - *Totale FBA*: il totale complessivo di unità presso Amazon, somma delle precedenti (disponibili + in transito + riservate).

Questa suddivisione dettagliata consente all'utente di capire non solo quanta merce è presente su Amazon, ma anche in quale stato logistico si trovi. La scelta di visualizzare separatamente queste voci è dunque motivata dal fornire un quadro più predittivo: il responsabile può stimare la copertura futura tenendo conto degli arrivi imminenti, non solo dello stock istantaneo.

Proseguendo, la dashboard include colonne relative alle vendite storiche dell'articolo, con un focus su orizzonti temporali significativi per il riordino. In particolare, vengono mostrati:

- Vendite ultimi N giorni: quantità vendute nell'ultimo mese, suddivise per canale (viene tipicamente distinto il totale vendite sul canale interno + FBM da un lato, e sul canale FBA dall'altro). Questa informazione fornisce la velocità di rotazione recente del prodotto.
- Vendite N giorni successivi anno precedente: per contestualizzare il dato di vendita recente, la dashboard mostra anche le vendite dei successivi n giorni dell'anno scorso per evidenziare trend stagionali imminenti. Questo confronto year-over-year aiuta a identificare pattern stagionali o crescite/declini della domanda: se l'anno scorso nel mese seguente le vendite erano molto più alte, l'utente può aspettarsi un aumento (stagionalità positiva) e regolarsi di conseguenza.

L'aggregazione dei dati di vendita è stata impostata su intervalli di 30, 60 e 90 giorni, con la possibilità per l'utente di modificare facilmente i periodi di riferimento e analizzare orizzonti temporali più brevi o più estesi. Questa configurazione rappresenta un compromesso efficace: i 30 giorni consentono di cogliere l'andamento recente, filtrando le oscillazioni settimanali troppo contingenti e offrendo al tempo stesso un margine di reazione utile per evitare l'esaurimento delle scorte. Il confronto con i dati dello stesso periodo dell'anno precedente contribuisce a ridurre il rischio di interpretazioni distorte dovute a picchi anomali, come promozioni o eventi stagionali, poiché introduce un solido riferimento storico. Nella dashboard, queste informazioni vengono presentate in forma numerica e compatta, accompagnate dalla variazione rispetto all'anno precedente, così che l'utente possa percepire immediatamente se la domanda di un determinato prodotto è in crescita o in calo rispetto al passato.

Un'altra colonna importante presente nell'interfaccia è quella dedicata alla “Scorta minima”. Questo campo rappresenta il livello di stock di sicurezza o riordino minimo impostato per l'articolo. In fase di progettazione, si è deciso di rendere visibile e modificabile tale parametro direttamente dalla dashboard: l'utente, tipicamente il buyer, può impostare o aggiornare la scorta minima desiderata per ciascun prodotto, in base alla sua esperienza o a politiche aziendali. La scorta minima funge da soglia di allerta: quando le giacenze effettive scendono al di sotto di questo valore, il sistema segnala un potenziale stock critico. Rendendo la *scorta minima* regolabile dall'utente, il sistema guadagna flessibilità: ad esempio, se per un prodotto strategico si vuole tenere più margine, il buyer può alzare la soglia indicante la scorta minima cosicché la dashboard, al successivo aggiornamento, presenterà questa nuova soglia segnalando prima la necessità di riordino. La dashboard offre questa funzionalità di input semplice, tramite un campo editabile, mantenendo però il controllo e la conferma esplicita da parte dell'utente per evitare modifiche accidentali.

La colonna cruciale dell'interfaccia è quella dell’“Alert”, che riassume la raccomandazione operativa del sistema per quel prodotto. Come anticipato, essa può assumere valori quali “OK”, “Riordinare”, “Trasferire” oppure “Trasferire!!” . La logica di visualizzazione prevede che questa colonna sia ben evidenziata, sia tramite colore di sfondo dell'intera riga, sia con testo formattato in modo chiaro (ad es. grassetto colorato). In questo modo, scorrendo la lista, l'utente vede immediatamente lo “stato” di ciascun SKU. Inoltre, la dashboard ordina di default i prodotti in base a questa colonna di alert, mettendo in cima quelli dove è richiesta

un’azione (in qualsiasi ordine di urgenza) e lasciando più in basso quelli “OK”. All’interno delle azioni richieste, è possibile che l’ordinamento secondario sia per urgenza o impatto (ad esempio, prima i “Trasferire!!” urgenti, poi i “Riordinare” e infine i “Trasferire” normali), cosicché l’elemento più critico in assoluto compaia in prima riga. Questo ordinamento è comunque personalizzabile: l’utente ha la facoltà di ordinare la tabella per altre colonne, ad esempio per nome prodotto, per quantità di stock, per vendite, ecc., semplicemente cliccando sull’intestazione della colonna desiderata (l’interfaccia usa la funzionalità di ordinamento client-side tipica delle tabelle interattive). Tuttavia, l’ordinamento per criticità è quello predefinito poiché rispecchia la priorità di utilizzo dello strumento (prima le urgenze).

Un ulteriore elemento della logica di visualizzazione è la presenza di funzionalità interattive di dettaglio. La dashboard non si limita a mostrare valori aggregati, ma offre all’utente la possibilità di esplorare maggiormente i dati di un singolo prodotto quando necessario. Su ciascuna riga, sono disponibili dei pulsanti di azione o link: uno di questi è il pulsante “Forecast”. Cliccando su *Forecast* in corrispondenza di un prodotto, si apre un popup in cui viene indicata anche la quantità consigliata da riordinare se l’articolo è in stato critico. Il sistema infatti, sfruttando l’algoritmo di forecasting sviluppato, è in grado di stimare non solo *se* un prodotto va riordinato ma anche *quanto* riordinare. In questo modo, l’utente riceve un supporto decisionale attivo: non deve dedurre manualmente la quantità dai numeri, perché il sistema la calcola e la esplicita. Questa funzionalità risponde a un preciso requisito di progetto: fornire azioni quantificate, non solo indicazioni generiche. Riassumendo la logica di visualizzazione, la dashboard mostra in forma tabellare tutti gli SKU rilevanti, combinando dati di stock, dati di vendita, parametri di riferimento (scorta minima, storico anno precedente) e risultati dei modelli analitici (stato di alert, previsioni). L’informazione è strutturata per colonne tematiche e arricchita da elementi interattivi (colori, ordinamento, filtri, popup di dettaglio) che ne facilitano l’interpretazione. Questa organizzazione permette all’utente di passare da una visione generale (panoramica di tutti i prodotti con i rispettivi stati) a una visione particolare (dettaglio del singolo prodotto) in pochi clic, mantenendo sempre il contesto. La scelta di quali campi mostrare e come presentarli è stata determinata dall’esperienza in azienda, ogni dato presente in dashboard risponde a una precisa domanda decisionale che il buyer o il responsabile della logistica si pone. Ad esempio: “Quanta scorta ho ora e dove si trova?”, “Quanto sto vendendo di questo prodotto? È una vendita anomala o stagionale?”, “Da quanto tempo non riassortisco questo articolo?”, “Rischio di finire le scorte prima che arrivi il prossimo riordino?”. La dashboard, tramite la sua logica di visualizzazione,

fornisce gli elementi per rispondere a tutte queste domande in modo semplice, evitando di dover incrociare manualmente informazioni da fonti diverse. In definitiva, la progettazione della visualizzazione è stata cruciale per trasformare la mole di dati e risultati prodotti dal sistema in conoscenza immediatamente fruibile per gli utenti finali.

5.4 Integrazione con i processi aziendali

L'introduzione della dashboard di monitoraggio in Igml non è un evento isolato dal contesto operativo, ma si inserisce in maniera organica nei processi aziendali esistenti, trasformandoli e ottimizzandoli. In questa sezione si descrive come l'interfaccia sia stata integrata nei flussi di lavoro quotidiani dell'azienda e quali vantaggi pratici ne derivano, evidenziando il passaggio da una gestione tradizionale delle scorte a una gestione data-driven supportata dal nuovo sistema.

Dal punto di vista tecnico, l'integrazione con le fonti dati aziendali è continua e automatizzata. Come descritto in precedenza, ogni giorno il sistema acquisisce gli aggiornamenti sulle giacenze e sulle vendite: ad esempio, durante le ore notturne il gestionale ERP esporta dei file CSV con le giacenze aggiornate del magazzino centrale e con le vendite del giorno, mentre tramite le API di Amazon si rilevano i cambiamenti di stock FBA, inclusi nuovi ordini, variazioni di riserva e arrivi di merce in transito. Questi dati vengono caricati nel database BI e su di essi vengono eseguiti gli algoritmi di classificazione e forecasting. Ogni mattina, pertanto, la dashboard rispecchia le ultime informazioni disponibili: le quantità in stock sono aggiornate al giorno precedente, le vendite recenti includono fino all'ultimo giorno chiuso, e le raccomandazioni (Riordino/Trasferimento) tengono conto degli ultimissimi trend. Per l'azienda ciò significa che il processo decisionale può avvenire su base giornaliera e con dati sempre aggiornati, anziché settimanale o mensile come avveniva in precedenza con i controlli manuali. In contesti di e-commerce, dove la domanda può variare rapidamente, questa prontezza rappresenta un netto miglioramento.

Il workflow operativo si svolge ora in maniera più strutturata intorno alla dashboard. Tipicamente, al mattino il buyer e il responsabile di magazzino aprono la dashboard per una rapida revisione dello stato delle scorte. In pochi secondi, grazie al popup che informa delle criticità, identificano gli articoli che richiedono attenzione. A questo punto il buyer annota quali prodotti sono in rosso e quindi da ordinare, per ciascuno di essi poi, analizza i dettagli

forniti (es. vendite ultimi 30 giorni, scorta attuale, arrivi già ordinati se visibili, ecc.) e soprattutto verifica, tramite il popup di forecast, quanto va riordinato. Sulla base delle informazioni disponibili, il buyer può elaborare un piano di approvvigionamento strutturato. Una volta individuato un prodotto da riordinare, l'analisi si estende spesso all'intero brand, al fine di valutare se includere ulteriori articoli all'interno dello stesso ordine. Questa prassi consente non solo di accrescere il potere negoziale con il fornitore, ottenendo condizioni economiche più vantaggiose, ma anche di soddisfare eventuali vincoli di ordine minimo richiesti, garantendo al contempo una maggiore efficienza complessiva nel processo di acquisto.. Prima della dashboard, questo lavoro veniva fatto attraverso l'aggregazione manuale di report dal gestionale e da Amazon, spesso con rischio di errore o di dimenticare qualche SKU, ora il buyer può affidarsi all'analisi del sistema per non trascurare nulla: sa che tutti i prodotti critici sono elencati e che per ognuno c'è un suggerimento quantitativo. Una volta decise le quantità, il buyer procede con le normali procedure d'ordine (es. inviando l'ordine al fornitore via ERP o email), ma in modo molto più mirato e giustificato da dati oggettivi.

Il responsabile di magazzino si focalizza sugli articoli evidenziati in blu “*Trasferire*”: verifica quanti pezzi sarebbero da inviare ai magazzini Amazon, confrontando lo stock centrale disponibile con la domanda su Amazon. Per esempio, se un prodotto ha 100 unità in magazzino centrale e ha venduto 50 pezzi nell'ultimo mese su Amazon (con tendenza in crescita), il responsabile potrebbe decidere, in accordo col buyer, di spedire magari 40-50 unità verso FBA, tenendo comunque una parte a magazzino centrale per gli ordini diretti. Il sistema facilita questa decisione mostrando sia il venduto recente su Amazon sia l'eventuale *differenza* rispetto all'anno precedente, indicatore di crescita. Così il responsabile può anche stimare se la domanda sta accelerando o se è costante. Una volta definito il piano di trasferimento, il responsabile di magazzino attiva il processo logistico: prepara la spedizione, genera le etichette e i documenti per Amazon FBA e affida la merce al corriere. Anche in questo caso, rispetto al passato, la differenza è notevole: precedentemente, i trasferimenti a FBA venivano fatti in modo reattivo, magari quando Amazon andava out-of-stock su un prodotto e ci si accorgeva in ritardo; ora invece si agisce proattivamente, spedendo prima che il prodotto si esaurisca su Amazon, evitando perdite di vendite e penalizzazioni di visibilità.

Una volta completate queste azioni, i due ruoli possono segnare internamente di aver preso in carico le raccomandazioni.

L'impatto sui processi decisionali interni è significativo. Si passa da una gestione reattiva (intervenire quando il problema, stock-out o overstock si è già manifestato o è imminente) a una gestione proattiva e pianificata. In passato, le criticità venivano spesso rilevate manualmente, attraverso controlli periodici a intervalli fissi, settimanali o quindicinali, sui livelli di scorta di ciascun magazzino. Ciò comportava che tra un controllo e l'altro potessero sfuggire situazioni di esaurimento, specialmente se le vendite andavo a registrare un improvviso aumento. Con la dashboard, invece, quotidianamente si effettua di fatto un controllo completo ed intelligente su tutti gli SKU, grazie all'automazione degli algoritmi. Questo riduce il carico di lavoro manuale e abbassa il rischio di rotture di stock non rilevate.

Dal punto di vista organizzativo, l'adozione della dashboard ha favorito una maggiore consapevolezza data-driven nel team. Le decisioni di riordino e trasferimento, che prima potevano essere basate in parte su intuizione o esperienza personale, ora vengono giustificate dai dati. Ciò, non solo aumenta la qualità delle decisioni, ma rende anche più facile comunicare e giustificare certe scelte ai livelli manageriali: il buyer può motivare un grosso ordine straordinario mostrando che il sistema ha previsto un picco di domanda (magari in base a trend storici o stagionalità), oppure il responsabile logistica può dimostrare la necessità di trasferire un certo numero di pezzi a Amazon spiegando che la Buy Box è detenuta e le vendite su quel canale stanno accelerando, quindi è un'opportunità di revenue da cogliere. In sintesi, le decisioni diventano tracciabili e spiegabili: ogni alert in dashboard è frutto di parametri oggettivi, e l'utente ha a disposizione quei parametri (vendite, stock, ranking, ecc.) per comprenderne il perché.

In conclusione, la dashboard implementata in Flask non è semplicemente un elemento tecnico, ma si configura come un vero e proprio strumento di gestione integrato nel workflow dell'azienda. L'introduzione di tale sistema ha ridisegnato parte dei processi di supply chain interni, introducendo automazione e intelligence laddove prima c'erano interventi manuali e statici. I risultati attesi in termini di efficienza, riduzione costi e aumento del servizio gettano le basi per un vantaggio competitivo nell'attuale mercato e-commerce, dove chi sa gestire meglio le proprie scorte può offrire un assortimento più affidabile e pronto alle richieste dei clienti. Nel capitolo successivo saranno discussi i risultati quantitativi di questa implementazione e i benefici misurati, ma già sul piano operativo è evidente come l'integrazione della dashboard abbia portato IgIm Store verso una gestione delle scorte più

moderna, proattiva e data-driven, in linea con le migliori pratiche di Business Intelligence applicate alla logistica.

Capitolo 6 - Validazione e risultati

6.1 Test su dataset storici e simulazioni

Per verificare l'efficacia delle soluzioni proposte, i modelli sono stati sottoposti a rigorosi test impiegando dati storici e simulazioni controllate. In particolare, si è proceduto a dividere il dataset in dati di training e di test: una porzione iniziale è stata utilizzata per addestrare i modelli e la restante parte per valutarne la capacità predittiva. La suddivisione è avvenuta in maniera stratificata rispetto alla variabile di output, così da preservare la proporzione originale delle classi anche nel test set. In questo modo, il test set contiene esempi di tutte le classi in proporzione simile al dataset complessivo, assicurando che la valutazione tenga conto della sbilanciamento delle classi presente nel problema.

Nel caso del modello di classificazione, dopo l'addestramento su dati storici, le sue previsioni sono state confrontate con le etichette reali. La valutazione è stata effettuata tramite metriche standard come l'accuratezza, la precisione, il recall e l'F1-score per ciascuna classe. Si è inoltre analizzata la *confusion matrix* per comprendere in quali casi il modello commetta errori e se tali errori siano accettabili dal punto di vista gestionale. Ad esempio, è stata posta particolare attenzione a capire se il modello tende a non segnalare elementi critici (falsi negativi per le classi *Riordinare* o *Trasferire*) oppure a generare falsi allarmi (falsi positivi di *Riordinare/Trasferire* su prodotti in realtà OK). Questa analisi qualitativa è cruciale per valutare l'impatto effettivo del sistema: un modello che manchi di segnalare un riordino urgente potrebbe causare rotture di stock significative, mentre qualche falso allarme di riordino è generalmente meno dannoso.

Oltre al semplice hold-out (suddivisione singola in training/test), sono state condotte validazioni incrociate (cross-validation) e simulazioni aggiuntive per rafforzare la robustezza dei risultati. Ad esempio, per i modelli di classificazione si è utilizzata una validazione k-fold stratificata durante la fase di training allo scopo di stimare le prestazioni medie e ottimizzare gli iperparametri. Questo procedimento ha permesso di simulare più scenari di suddivisione dei dati, verificando che i risultati ottenuti non fossero casualmente favorevoli o sfavorevoli a seconda di una particolare partizione dei dati. Per il modello di forecasting SARIMA, a differenza della classificazione, che ha potuto contare su un dataset sufficientemente ricco per test e validazioni incrociate, l'algoritmo di forecasting sviluppato si trova ancora in una fase embrionale. La limitata disponibilità di serie storiche circa un solo anno di dati mensili non ha permesso di effettuare un vero e proprio backtesting su più stagioni, come sarebbe auspicabile

in un modello di previsione della domanda. Per questa ragione, invece di condurre test quantitativi su dati passati, si è scelto di adottare un approccio esplorativo basato sul confronto con la realtà operativa: le previsioni prodotte dal modello SARIMA sono state affiancate agli ordini effettuati manualmente dai buyer, così da valutare in che misura le stime dell'algoritmo si avvicinano alle decisioni effettivamente prese in azienda.

Questo metodo di validazione “sul campo” ha consentito di osservare punti di contatto e differenze: in diversi casi il modello ha suggerito quantità comparabili a quelle riordinate dai buyer, mentre in altri ha fornito valori divergenti, soprattutto per SKU caratterizzati da domanda molto irregolare. Queste discrepanze non vanno interpretate come errori definitivi, bensì come indicazioni utili per migliorare il modello. Esse hanno messo in luce i limiti di un dataset ancora breve e la necessità di arricchire l’algoritmo con variabili esogene e con orizzonti temporali più ampi. Complessivamente, l’uso combinato di test su dati storici reali e simulazioni artificiali ha fornito un quadro completo della validità e affidabilità dei modelli sviluppati, prima di procedere alla loro implementazione effettiva.

6.2 Analisi delle performance degli algoritmi

In questa sezione vengono presentati i risultati quantitativi ottenuti dai modelli sviluppati. Durante le fasi preliminari del progetto sono stati testati anche altri algoritmi di classificazione, come la regressione logistica multiclasse e il k-Nearest Neighbors. Tuttavia, sia l’analisi della letteratura di riferimento, sia le evidenze emerse dai test iniziali, che hanno mostrato prestazioni sensibilmente inferiori, hanno portato a escludere tali metodi dal percorso principale della ricerca. Per questa ragione, l’attenzione è stata focalizzata unicamente su tre modelli ritenuti più promettenti e solidi: Decision Tree, Random Forest e XGBoost, che vengono qui confrontati e analizzati in dettaglio. Quindi si procede dapprima ad analizzare le performance del modello di classificazione, confrontando i tre algoritmi considerati soffermandosi sull’impatto delle tecniche di bilanciamento (SMOTEENN) sui risultati. Successivamente, si illustrano i risultati del modello di forecasting (SARIMA), evidenziando la capacità di previsione sulle serie storiche di vendita. Le metriche principali utilizzate per valutare i modelli di classificazione includono l’accuratezza complessiva e gli F1-score per ciascuna classe, con particolare enfasi sulle classi *Riordinare* e *Trasferire* che sono quelle di maggior interesse. Per il modello di forecasting si considerano invece misure di errore sulle previsioni e la capacità di cogliere trend/stagionalità.

6.3 Risultati dei modelli di classificazione

Nei successivi paragrafi si andranno ad analizzare il funzionamento dei singoli modelli sul dataset, durante l'analisi dei risultati si dovrà tenere conto che la creazione di un sistema di classificazione per la gestione delle scorte diventa più complicata quando le classi si presentano sbilanciate. Nel dataset di IGLM.Store, la maggioranza delle etichette appartiene alla classe OK, parametro che indica la necessità di compiere nessuna azione. Le classi Riordinare, Trasferire e soprattutto Trasferire!! risultano invece significativamente sottorappresentate, sebbene siano proprio queste a rivestire il ruolo più critico dal punto di vista operativo.

In scenari di questo tipo, i modelli di classificazione tendono a privilegiare la classe più frequente, massimizzando l'accuratezza complessiva ma trascurando le minoranze. Ciò genera una conseguenza diretta e potenzialmente grave: i casi di effettiva criticità, come la necessità di trasferire prodotti verso Amazon FBA o di segnalarne l'urgenza massima, vengono classificati come OK e dunque ignorati dal sistema. Per un contesto aziendale in cui la tempestività e la correttezza delle decisioni incidono direttamente sul rischio di stock-out e sui costi di magazzino, questa distorsione rappresenta un limite inaccettabile.

Per ovviare a tale problema, nella fase di addestramento è stata applicata la tecnica SMOTEENN, Synthetic Minority Over-sampling Technique + Edited Nearest Neighbors, che combina due strategie complementari. Da un lato, SMOTE genera nuovi campioni sintetici per le classi meno rappresentate, interpolando gli esempi esistenti e ampliando lo spazio decisionale a loro favore. Dall'altro lato, ENN rimuove i campioni rumorosi o ambigui della classe maggioritaria, in particolare quelli situati nelle zone di confine con le classi minori. Il risultato è un dataset più equilibrato e pulito, che consente agli algoritmi di apprendere pattern associati alle classi critiche senza esserne sopraffatti dal peso della classe dominante.

L'applicazione di SMOTEENN comporta inevitabilmente un compromesso. Da un lato, aumenta il recall delle classi minoritarie, migliorando la capacità del modello di intercettare correttamente i casi di Riordinare e Trasferire. Dall'altro, può determinare una leggera riduzione della precisione, introducendo falsi positivi: alcuni prodotti in realtà corretti, vengono classificati erroneamente come da riordinare o da trasferire. Tuttavia, nel contesto specifico di IGLM.Store, questa scelta metodologica è risultata coerente con le priorità aziendali. Un falso positivo, infatti, può comportare al massimo un ordine anticipato o un

trasferimento non strettamente necessario, con un impatto economico contenuto. Un falso negativo, al contrario, rischia di generare un mancato riordino tempestivo o la mancata riallocazione di stock, con conseguente perdita di vendite, peggioramento del posizionamento su Amazon e aumento dei costi operativi.

L'introduzione di SMOTEENN rappresenta quindi un passaggio fondamentale per garantire che i modelli sviluppati non si limitino a replicare l'inerzia statistica dei dati, ma offrano un reale valore aggiunto alle decisioni operative. Nelle sezioni successive verranno analizzati i risultati ottenuti dai singoli modelli di classificazione, prima nella loro configurazione originale e poi nella versione bilanciata, così da evidenziare l'efficacia e i limiti di entrambe le impostazioni. Successivamente, i modelli bilanciati verranno confrontati tra loro in modo sistematico e verrà determinato quello che, per accuratezza, capacità di richiamo e robustezza complessiva, risulta più adatto a essere implementato come strumento operativo a supporto dei buyer di IGLM.Store.

6.3.1 Decision Tree

Il modello Decision Tree rappresenta una delle tecniche di classificazione più semplici e interpretabili disponibili nel panorama del machine learning. La sua logica si basa sulla costruzione di un albero di decisione in cui, ad ogni nodo, il dataset viene suddiviso sulla base di una condizione ottimale scelta per massimizzare la purezza delle classi nei nodi figli. Questa caratteristica conferisce al modello un notevole grado di trasparenza, in quanto consente di ricostruire facilmente il percorso seguito per classificare ogni osservazione. Proprio per questa sua natura intuitiva, il Decision Tree è stato scelto come punto di partenza per il progetto, con l'obiettivo di ottenere una baseline iniziale e comprendere meglio la distribuzione dei dati e le relazioni fra le variabili più significative.

In particolare, il modello di classificazione Decision Tree, che è stato implementato, applica:

- `random_state=1`, un parametro opzionale utilizzato per riprodurre i risultati in modo deterministico. Impostando questo parametro su un valore fisso, si garantisce che il modello venga addestrato nello stesso modo ogni volta che il codice viene eseguito. In questo caso, il valore 1 è stato scelto casualmente come seme casuale per la riproducibilità dei risultati.

- `p_grid = {"max_depth": [2, 3, 4, 5, 10, 15, 20]}` definisce una griglia di parametri (`p_grid`) che specifica i valori che il parametro `max_depth` del classificatore decision tree deve assumere durante la ricerca dei migliori parametri tramite `GridSearchCV`. In particolare, la griglia di parametri specifica che il parametro `max_depth` può assumere i valori [2, 3, 4, 5, 10, 15 ,20]. Durante la ricerca dei migliori parametri, `GridSearchCV` testerà il classificatore decision tree con ogni valore possibile di `max_depth` all'interno della griglia di parametri specificata e restituirà la combinazione di parametri che produce la migliore performance sul dataset di validazione.
- `inner_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)` definisce una validazione incrociata stratificata con 5 fold. La validazione incrociata è una tecnica utilizzata per valutare le prestazioni di un modello di machine learning. In particolare, la validazione incrociata suddivide il set di dati in k fold (in questo caso, k = 5), utilizzando uno dei fold come set di test e gli altri k-1 fold come set di addestramento. Questa operazione viene ripetuta k volte, in modo che ogni fold sia utilizzato come set di test una volta. La validazione incrociata stratificata garantisce una distribuzione delle classi all'interno dei fold simile alla distribuzione delle classi nell'intero set di dati. Ciò è particolarmente utile quando si lavora con set di dati sbilanciati, in cui una classe è rappresentata da un numero significativamente inferiore di campioni rispetto alle altre classi. Il parametro `shuffle=True` indica che i dati verranno mischiati prima della suddivisione in fold, mentre il parametro `random_state=1` garantisce la riproducibilità dei risultati.
- `clf = GridSearchCV(estimator=dt, param_grid=p_grid, cv=inner_cv, verbose=0)` crea un oggetto `GridSearchCV` che viene utilizzato per eseguire una ricerca a griglia per trovare i migliori iperparametri per il modello di classificazione basato su albero decisionale, utilizzando una cross-validation stratificata (`inner_cv`) con 5 fold. Il parametro `param_grid` specifica un dizionario di iperparametri (`max_depth`) da testare durante la ricerca a griglia. La `GridSearchCV` itera su tutte le possibili combinazioni di iperparametri specificati in `param_grid`, addestra un modello con ciascuna combinazione e valuta le prestazioni del modello con la cross-validation.

In una prima fase, l'algoritmo è stato addestrato sul dataset non bilanciato, cioè nella sua forma originaria, caratterizzata da una forte predominanza della classe OK, che rappresenta la maggioranza assoluta dei casi. In questo scenario, il Decision Tree ha ottenuto un'accuratezza complessiva pari a 0,86, un valore apparentemente elevato ma fortemente influenzato dallo

sbilanciamento. Analizzando nel dettaglio le singole classi, si osserva che il modello riesce a classificare correttamente la grande maggioranza dei prodotti appartenenti alla classe OK (recall pari a 0,93), mostrando una discreta affidabilità anche sulla classe Riordinare (recall 0,74). Le criticità emergono però chiaramente sulle classi meno rappresentate, quelle di maggiore interesse operativo: il recall per Trasferire si ferma infatti a 0,31, mentre per la classe più rara, Trasferire!!, il modello non è in grado di identificare alcun caso (recall 0,00). Ciò si traduce in un macro F1-score di 0,50, valore che riflette uno squilibrio significativo fra le diverse classi. In altre parole, l'albero tende a classificare quasi tutto come OK o Riordinare, trascurando i casi in cui sarebbe necessaria un'azione immediata, generando così falsi negativi altamente rischiosi per la gestione delle scorte.

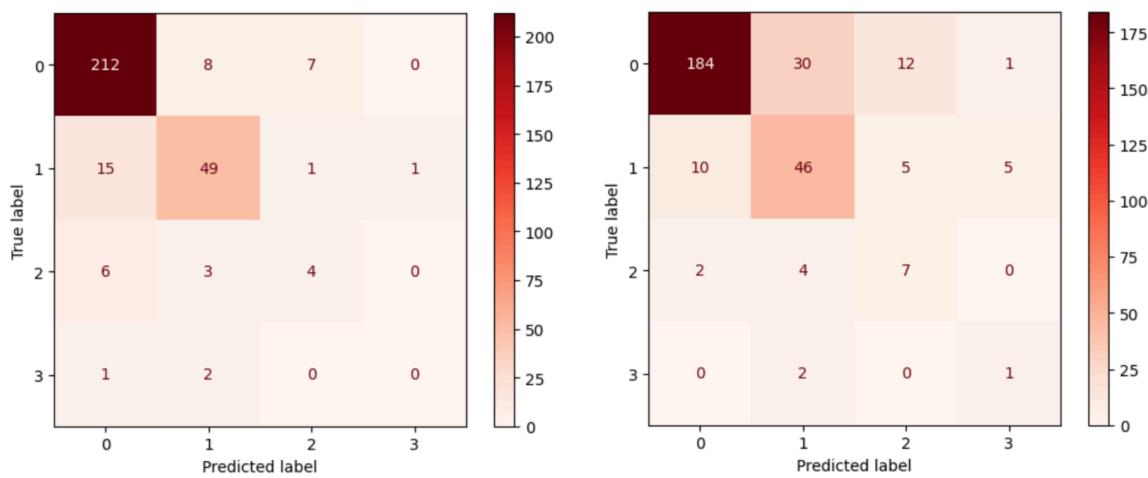
Questa tendenza è coerente con la natura del Decision Tree e con la struttura dei dati a disposizione: trattandosi di un modello che ottimizza localmente le soglie di separazione per massimizzare l'accuratezza complessiva, esso privilegia inevitabilmente la classe più frequente, con l'effetto collaterale di ridurre la capacità predittiva sulle classi minoritarie. In un contesto aziendale, ciò equivale a non intercettare situazioni critiche di trasferimento di stock, che rappresentano invece le condizioni più delicate da gestire.

Per migliorare la capacità del modello di riconoscere anche le classi meno rappresentate, si è proceduto ad applicare la tecnica di bilanciamento SMOTEENN in fase di training. Tale approccio, come già discusso in §6.3.3, combina due operazioni: da un lato la generazione di nuovi campioni sintetici per le classi minoritarie (SMOTE), dall'altro l'eliminazione di esempi rumorosi della classe maggioritaria (ENN). L'effetto complessivo è la costruzione di un dataset di addestramento più equilibrato, in cui le classi di interesse non sono più sommerse dal volume di esempi OK.

Con il dataset bilanciato, il Decision Tree ha mostrato un comportamento diverso. L'accuratezza complessiva si è ridotta a 0,77, valore inferiore rispetto al modello non bilanciato, ma il calo è compensato da un significativo incremento delle metriche sulle classi minoritarie. Il recall per Riordinare si è mantenuto elevato (0,70), mentre quello per Trasferire è quasi raddoppiato, passando da 0,31 a 0,54. Per la classe Trasferire!!, inesistente nella predizione del modello sbilanciato, si registra finalmente un miglioramento, con un recall pari a 0,33. Il macro F1-score cresce leggermente, da 0,50 a 0,52, segno che il modello ha

acquisito una maggiore sensibilità complessiva, pur a fronte di una diminuzione dell'accuratezza globale.

In sostanza, il bilanciamento ha reso il Decision Tree più “cauto”, riducendo la tendenza a classificare in modo conservativo i casi incerti come OK. Di conseguenza, il modello produce più falsi positivi (cioè segnala come critici anche prodotti che non lo sono), ma riduce in maniera importante i falsi negativi sulle classi di maggiore rilevanza gestionale. Dal punto di vista aziendale, questo compromesso risulta accettabile: un falso positivo si traduce al massimo in un ordine o trasferimento anticipato non strettamente necessario, mentre un falso negativo può comportare uno stock-out o una mancata disponibilità su Amazon, con conseguente perdita di vendite e peggioramento del ranking del prodotto.



6.1 Confusion Matrix Decision Tree non bilanciato e bilanciato

In conclusione, il Decision Tree conferma il proprio valore come baseline interpretabile, utile per comprendere le logiche di classificazione e per avere un primo riferimento prestazionale. Tuttavia, le sue performance limitate, soprattutto sulle classi Trasferire e Trasferire!!, ne riducono l'idoneità come unico modello di supporto alle decisioni. Il suo ruolo rimane dunque quello di punto di partenza, ma i risultati ottenuti motivano la necessità di ricorrere a metodi più robusti e flessibili, come gli ensemble, capaci di catturare meglio la complessità e la variabilità del fenomeno analizzato.

6.3.2 Random Forest

La Random Forest rappresenta un'evoluzione naturale del Decision Tree e si basa su un approccio ensemble, in cui centinaia di alberi decisionali vengono addestrati su differenti sottocampioni dei dati e delle feature, e le loro predizioni combinate mediante voto maggioritario. Questo metodo consente di mitigare i limiti di un singolo albero, riducendo sensibilmente il rischio di overfitting e migliorando la capacità di generalizzazione. Nel contesto del progetto Igml Store, la Random Forest è stata valutata sia in versione non bilanciata che con dataset bilanciato tramite SMOTEENN. I risultati che seguiranno sono stati ottenuti con i seguenti parametri:

- n_estimators=20: specifica il numero di alberi decisionali (o estimatori) che verranno creati nella foresta casuale. In questo caso, verranno creati 20 alberi di decisione.
- random_state=1: determina il seme casuale che viene utilizzato per riprodurre gli stessi risultati ogni volta che il codice viene eseguito.
- class_weight="balanced": un parametro opzionale che specifica il peso da assegnare alle classi durante l'addestramento del modello. La stringa "balanced" indica che il peso delle classi verrà assegnato in modo tale da bilanciare il numero di campioni in ogni classe.

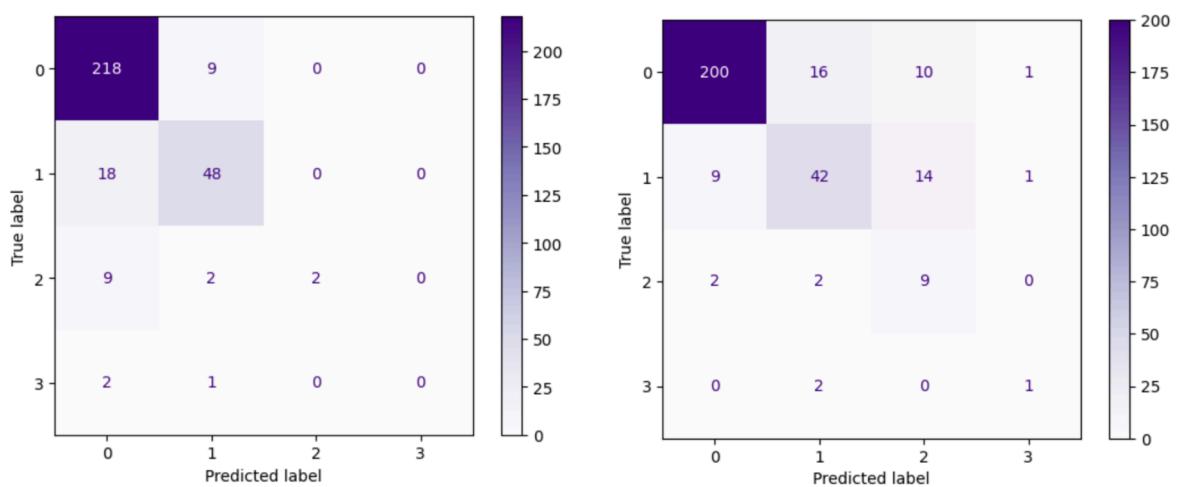
Nella configurazione non bilanciata, il modello ha raggiunto un'accuratezza globale pari a 0,87, superiore a quella del Decision Tree. L'analisi delle metriche per classe mostra un recall eccellente per OK (0,96) e buono per Riordinare (0,73), ma ancora problematico per le classi meno frequenti: il recall per Trasferire si attesta a 0,15, mentre per Trasferire!! rimane nullo (0,00). Il macro F1-score complessivo è pari a 0,46, leggermente inferiore rispetto al Decision Tree non bilanciato. Questo risultato, apparentemente controiduitivo, dipende dal fatto che la Random Forest, pur offrendo ottime performance sulle classi più comuni, tende comunque a trascurare le classi critiche con pochi esempi, riducendo l'equilibrio globale del modello. In altre parole, il voto aggregato di molti alberi amplifica la preferenza per le classi maggioritarie, lasciando in ombra i pattern delle classi più rare.

Con l'applicazione della tecnica di bilanciamento SMOTEENN, le prestazioni cambiano in modo significativo. L'accuratezza complessiva scende a 0,78, come conseguenza naturale della maggiore attenzione alle classi minoritarie. Tuttavia, il recall per Riordinare aumenta notevolmente, raggiungendo 0,92, e il recall per Trasferire cresce da 0,15 a 0,67. Per la classe

Trasferire!!, inesistente nelle predizioni del modello non bilanciato, si ottiene finalmente un riconoscimento parziale con recall pari a 0,33. Anche in questo caso, si osserva una lieve riduzione della precisione sulle classi di interesse, ma ciò è compensato da un sostanziale miglioramento della capacità del modello di individuare correttamente i casi critici. Il macro F1-score passa a 0,66, segnando un progresso significativo rispetto al valore di 0,46 del modello sbilanciato.

Dal punto di vista interpretativo, la Random Forest fornisce anche una misura dell'importanza delle feature, calcolata come riduzione media dell'impurità nei nodi degli alberi. I risultati mostrano che le variabili più determinanti sono lo stock disponibile in FBA, le vendite degli ultimi 90 giorni e lo stock FBM, seguite da parametri competitivi come il prezzo BuyBox e il ranking di categoria. Questa gerarchia è perfettamente coerente con le logiche operative: le decisioni di riordino e trasferimento dipendono principalmente dalla disponibilità immediata di stock e dalla velocità di rotazione del prodotto, integrate da fattori esterni che influenzano la competitività su Amazon.

Dal punto di vista computazionale, l'addestramento della Random Forest si è dimostrato efficiente, con tempi contenuti grazie al numero relativamente ridotto di osservazioni e feature. Inoltre, il modello si è rivelato stabile: variazioni limitate nei dati di input non hanno generato fluttuazioni marcate nelle predizioni, segno della robustezza derivante dall'aggregazione di molti alberi.



6.2 Confusion Matrix Random Forest non bilanciato e bilanciato

In conclusione, la Random Forest si conferma un modello altamente efficace per il problema analizzato. Sebbene la versione non bilanciata presenti buoni livelli di accuratezza, la sua incapacità di rilevare le classi più critiche ne limita l'utilità pratica. La versione bilanciata, invece, pur con un lieve sacrificio di precisione e accuratezza complessiva, garantisce una copertura quasi completa delle situazioni operative di interesse, trasformandosi così in una soluzione affidabile e adatta a supportare concretamente i buyer di IGLM.Store nelle decisioni di procurement.

6.3.3 XGBoost

L'algoritmo XGBoost (Extreme Gradient Boosting) è stato adottato con l'obiettivo di verificare se un approccio di boosting sequenziale potesse migliorare le prestazioni rispetto alla Random Forest, in particolare nella capacità di riconoscere le classi meno rappresentate. XGBoost è noto per l'elevata accuratezza e per la capacità di catturare pattern complessi grazie alla costruzione iterativa di alberi deboli, ciascuno dei quali corregge progressivamente gli errori commessi dai precedenti.

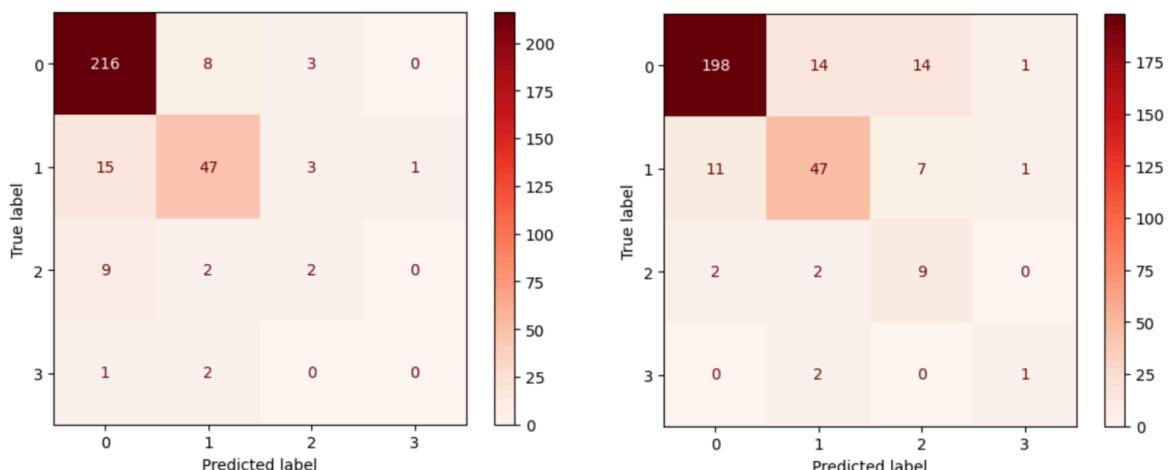
Gli iperparametri scelti per il test attraverso l'XGBoost sono:

- `max_depth = 3`: questo parametro fissa la profondità massima degli alberi decisionali utilizzati nel boosting. Una profondità ridotta limita l'overfitting.
- `n_estimators = 2000`: è il numero di alberi da costruire nella sequenza di boosting.
- `learning_rate = 0.2`: è il tasso di apprendimento, che determina quanto ogni nuovo albero corregge gli errori dei precedenti.

Nella configurazione non bilanciata, il modello ha raggiunto un'accuratezza complessiva pari a 0,86, un valore elevato che tuttavia nasconde una forte disparità tra le classi. La classe OK è stata identificata con ottima affidabilità, con un livello di richiamo molto vicino alla perfezione, mentre la classe Riordinare ha mostrato performance ancora accettabili, con valori di precisione e recall in linea con le attese. La situazione si complica, invece, per le classi più rare: Trasferire presenta un richiamo molto basso e un F1-score insufficiente, mentre la classe Trasferire!! non è stata mai riconosciuta, con valori nulli su tutte le metriche. Il macro F1-score complessivo si è quindi attestato a 0,47, evidenziando uno squilibrio marcato e una scarsa capacità del modello di generalizzare sulle situazioni più critiche.

Per ridurre questo limite è stata applicata la tecnica di bilanciamento SMOTEENN, che ha modificato sensibilmente il comportamento del modello. L'accuratezza complessiva è leggermente scesa a 0,83, ma le metriche per le classi minoritarie hanno registrato un miglioramento netto. La classe Trasferire, che nella configurazione sbilanciata mostrava un richiamo pari a 0,23, ha raggiunto un valore di 0,69, con un F1-score passato da 0,27 a 0,43. Anche la classe Trasferire!! ha iniziato a essere parzialmente riconosciuta, con valori di richiamo e precisione pari a 0,33, mentre la classe Riordinare ha mantenuto risultati stabili, confermando che il bilanciamento non ha compromesso le sue prestazioni. La classe OK ha subito una lieve riduzione del richiamo, pur mantenendo valori elevati che ne confermano la robustezza. Nel complesso, il macro F1-score è salito a 0,59, segnalando una maggiore equità del modello nella gestione delle diverse categorie e una sensibilità rafforzata verso gli scenari critici.

Il confronto fra le due configurazioni mostra come XGBoost, se addestrato su dati non bilanciati, tenda a privilegiare in modo marcato le classi maggioritarie, trascurando completamente quelle più rilevanti dal punto di vista operativo. L'applicazione di SMOTEENN consente invece di recuperare gran parte delle situazioni di Trasferire e di intercettare almeno una quota dei casi più urgenti, accettando in cambio un incremento dei falsi positivi e una riduzione complessiva dell'accuratezza. Questo compromesso risulta coerente con gli obiettivi aziendali del progetto: in un contesto operativo come quello di IGLM.Store, è preferibile avere qualche allarme in eccesso piuttosto che mancare l'individuazione di condizioni che possono portare a stock-out o inefficienze gravi.



6.3 Confusion Matrix XGBoost non bilanciato e bilanciato

6.3.4 Confronto tra i modelli di classificazione

Il confronto complessivo tra i tre modelli di classificazione evidenzia differenze significative in termini di accuratezza globale, capacità di generalizzazione e soprattutto comportamento rispetto alle classi minoritarie. Il Decision Tree, pur rappresentando una soluzione interpretabile e immediata, ha mostrato prestazioni limitate. La sua struttura semplice lo porta infatti a privilegiare la classe OK, producendo un numero elevato di falsi negativi per le categorie Riordinare e Trasferire, e dimostrando una sensibilità pressoché nulla nei confronti della classe Trasferire!. Si tratta quindi di un modello utile come punto di partenza per comprendere la logica di classificazione, ma inadeguato per un utilizzo pratico all'interno di un sistema decisionale complesso.

La Random Forest ha evidenziato un salto qualitativo importante. Grazie alla natura ensemble, che aggrega i risultati di numerosi alberi decisionali, il modello si è rivelato molto più stabile e robusto. L'accuratezza complessiva è aumentata in maniera significativa e, soprattutto, le prestazioni sulle classi critiche sono migliorate sensibilmente. Il modello è riuscito a intercettare con buona affidabilità i casi Riordinare e Trasferire, dimostrando un equilibrio più solido tra precisione e richiamo. Inoltre, l'analisi dell'importanza delle variabili ha confermato la coerenza del comportamento predittivo con la logica di business, ponendo al centro la relazione tra stock disponibile, vendite recenti e posizionamento competitivo.

Il modello XGBoost, noto per la sua capacità di ottimizzare sequenzialmente gli errori residui, ha mostrato prestazioni globali molto vicine a quelle della Random Forest. L'accuratezza complessiva si è mantenuta su livelli simili, con alcuni vantaggi specifici sulla classe Trasferire, che in più di un'occasione è stata riconosciuta con maggiore sensibilità. Tuttavia, questo miglioramento si è accompagnato a un incremento dei falsi positivi, in particolare sulla classe Riordinare, con un conseguente abbassamento della precisione. XGBoost si conferma dunque un modello competitivo, ma leggermente più complesso da interpretare e meno stabile rispetto alla Random Forest in un dataset di dimensioni ridotte come quello a disposizione.

In sintesi, il confronto tra i tre modelli porta a considerare la Random Forest come la soluzione più equilibrata per il caso in esame. Essa rappresenta un compromesso ottimale tra accuratezza complessiva, capacità di individuare i casi critici e robustezza interpretativa. Pur riconoscendo i margini di miglioramento offerti da XGBoost e la semplicità didattica del Decision Tree, la Random Forest emerge come il modello più adatto all'integrazione nel

sistema sviluppato, fornendo un supporto decisionale affidabile e coerente con le esigenze operative di IGLM.Store.

6.4 Risultati del modello di forecasting (SARIMA)

Il modello SARIMA (Seasonal AutoRegressive Integrated Moving Average) è stato selezionato come strumento di partenza per affrontare il problema del forecasting della domanda. La scelta si è basata sulla sua capacità di rappresentare in modo relativamente semplice serie temporali che presentano sia una componente di trend sia una componente stagionale. In teoria, il SARIMA consente di modellare dinamiche ricorrenti, come i picchi di vendita legati a particolari periodi dell'anno, e di proiettare tali pattern nel futuro, fornendo previsioni interpretabili e accompagnate da intervalli di confidenza.

Nel caso di IGLM.Store, tuttavia, la base dati disponibile al momento dello sviluppo era molto limitata: circa 12–13 osservazioni mensili per ciascun prodotto. Questo ha reso impossibile una calibrazione approfondita dei parametri e, soprattutto, un backtesting statisticamente robusto. Per aggirare questa criticità, il modello è stato utilizzato in maniera sperimentale, con l'obiettivo principale di generare previsioni indicative e confrontarle con le scelte effettivamente compiute dai buyer nel processo di riordino.

Dal punto di vista tecnico, l'implementazione ha previsto una configurazione standard SARIMA(p,d,q)(P,D,Q) $_s$ con $s=12$ per cogliere eventuali ciclicità annuali. I valori dei parametri sono stati fissati in base a criteri euristici e alla letteratura di riferimento, senza un'ottimizzazione puntuale per ciascun SKU, proprio a causa della scarsità di dati storici. L'algoritmo è stato eseguito in batch su più serie, generando per ciascun prodotto una proiezione di vendite per i mesi successivi, tipicamente su un orizzonte pari al lead time di approvvigionamento sommato a un margine di sicurezza.

Il confronto con gli ordini manuali ha evidenziato alcuni aspetti significativi. Per prodotti con domanda relativamente stabile, le quantità suggerite dal modello si sono avvicinate a quelle effettivamente ordinate dai buyer, confermando la validità dell'approccio. In altri casi, soprattutto per SKU a domanda intermittente o soggetti a forti stockout, il modello ha mostrato previsioni meno affidabili, spesso sottostimando la domanda latente. In queste situazioni, l'esperienza del buyer ha continuato a giocare un ruolo determinante, integrando e correggendo le stime del sistema.

L'esperimento ha dunque confermato che, pur non essendo ancora uno strumento operativo maturo, SARIMA può rappresentare un valido supporto al processo decisionale. Il suo ruolo, in questa fase, è quello di fornire un segnale direzionale: evidenziare tendenze di crescita o calo della domanda e suggerire ordini di grandezza per i riordini. La decisione finale rimane in capo al buyer, che può bilanciare i suggerimenti quantitativi del modello con la conoscenza contestuale di mercato e di fornitura.

Per il futuro, sarà fondamentale estendere l'orizzonte dei dati storici, affinare la selezione dei parametri e integrare nel modello variabili esogene (come promozioni, ranking di vendita o indicatori di mercato). In questo modo sarà possibile trasformare l'attuale approccio sperimentale in un sistema predittivo solido, capace di sostenere in maniera autonoma e affidabile le politiche di procurement di IGLM.Store.

Capitolo 7 - Conclusioni

Il lavoro di ricerca e sviluppo presentato in questa tesi si è posto l’obiettivo di realizzare un sistema di supporto decisionale per la gestione delle scorte e dei riordini in IGLM.Store, combinando strumenti di Business Intelligence con algoritmi di Machine Learning. L’intero percorso ha avuto come filo conduttore l’introduzione di un approccio data-driven in un contesto aziendale complesso, caratterizzato da vendite multicanale e dalla necessità di bilanciare accuratamente disponibilità e costi di stock.

Sin dalle prime fasi, l’analisi ha mostrato come i metodi tradizionali, pur offrendo una base operativa, non fossero sufficienti a cogliere la dinamicità dei mercati digitali e le specificità di Amazon, con i suoi meccanismi competitivi legati alla Buy Box e alla gestione dei magazzini FBA. Da qui è nata la necessità di sviluppare un tool interno in grado di integrare dati provenienti da diverse fonti, elaborarli con algoritmi predittivi e proporre azioni concrete e tempestive ai buyer.

7.1 Risultati principali

I risultati ottenuti evidenziano progressi significativi rispetto alle pratiche precedenti. La Random Forest, individuata come modello più equilibrato tra accuratezza e interpretabilità, ha garantito performance stabili nella classificazione, riuscendo a riconoscere con elevata affidabilità i casi in cui era necessario riordinare o trasferire prodotti. L’integrazione del forecasting basato su SARIMA ha permesso di completare il sistema con la capacità di stimare non solo se intervenire, ma anche quanto ordinare.

L’elemento più rilevante, tuttavia, è che il sistema non è rimasto confinato alla dimensione sperimentale. Il tool è stato introdotto operativamente in azienda e, dopo alcuni mesi di utilizzo, i riscontri sul campo si sono dimostrati ottimi. In particolare, è stata osservata una riduzione significativa degli stockout, con conseguente miglioramento della disponibilità dei prodotti su Amazon e sull’eCommerce proprietario. Tale risultato conferma la validità dell’approccio proposto e ne dimostra l’impatto diretto sulle performance aziendali.

7.2 Limiti dello studio

Nonostante i risultati incoraggianti, il lavoro presenta alcune limitazioni che meritano di essere evidenziate. Durante la fase di preprocessing è stato necessario eliminare alcune feature

a causa della presenza di dati mancanti. Questa scelta, se da un lato ha permesso di mantenere la qualità del dataset utilizzato per l’addestramento, dall’altro ha comportato la perdita di informazioni potenzialmente rilevanti, che avrebbero potuto arricchire ulteriormente i modelli predittivi.

Inoltre, la disponibilità di serie storiche limitate ha reso difficile la piena validazione dei modelli di forecasting. Il SARIMA, applicato su dataset relativamente brevi, ha mostrato buone capacità di cogliere i trend generali, ma con limiti evidenti nella previsione di dinamiche stagionali o di lungo periodo. Queste criticità non compromettono la validità del sistema, ma ne riducono la capacità di generalizzare in scenari più complessi.

7.3 Sviluppi futuri

Gli sviluppi futuri del progetto dovranno concentrarsi su tre direzioni principali. In primo luogo, sarà fondamentale arricchire il dataset, riducendo l’incidenza dei dati mancanti e garantendo una base storica più estesa. Questo consentirà di migliorare la qualità sia della classificazione sia del forecasting, rendendo le previsioni più robuste e affidabili.

In secondo luogo, si sta valutando la possibilità di trasformare la classificazione attuale da un modello multiclasse a due classificazioni binarie distinte: *riordinare vs. ok* e *trasferire vs. ok*. Questa scelta metodologica potrebbe ridurre gli errori di confusione tra classi simili e aumentare la capacità predittiva dei modelli.

Infine, si prevede di ampliare ulteriormente l’integrazione dei dati, includendo variabili esogene, come trend di mercato, indicatori macroeconomici o andamenti stagionali per migliorare la qualità delle previsioni di domanda.

7.4 Contributo pratico per IGLM.Store

Uno degli aspetti più significativi di questo lavoro è l’impatto diretto che ha avuto su IGLM.Store. La tesi non si è limitata a produrre un prototipo accademico, ma ha fornito all’azienda uno strumento concreto, oggi utilizzato quotidianamente dai buyer per guidare le decisioni di riordino e trasferimento.

Il contributo è stato duplice: da un lato, si sono introdotti strumenti operativi che hanno migliorato l’efficienza dei processi, riducendo stockout e ottimizzando l’allocazione delle

scorte tra i diversi magazzini, dall'altro, si è favorito un cambiamento culturale all'interno dell'organizzazione, promuovendo l'adozione di un approccio basato sui dati. La possibilità di osservare in dashboard le indicazioni generate dagli algoritmi ha aumentato la consapevolezza del potenziale insito nell'analisi predittiva e ha facilitato il dialogo tra le diverse funzioni aziendali coinvolte nella supply chain.

7.5 Considerazioni finali

In conclusione, la tesi ha dimostrato la fattibilità e l'utilità di un sistema di Business Intelligence e Machine Learning applicato alla gestione delle scorte in un contesto e-commerce multicanale. I risultati conseguiti sul campo confermano che l'integrazione di strumenti predittivi nei processi decisionali può produrre benefici tangibili, sia in termini di efficienza operativa sia di vantaggio competitivo.

L'esperienza maturata con questo progetto apre la strada a ulteriori sviluppi e rappresenta un punto di partenza per consolidare la trasformazione digitale di IGLM.Store. La riduzione degli stockout, l'ottimizzazione dei riordini e l'introduzione di una cultura aziendale orientata ai dati, costituiscono i principali traguardi raggiunti, a testimonianza di come la ricerca accademica possa tradursi in valore reale per le imprese.

Bibliografia

- Akter, & Wamba. (2016). Big data analytics in e-commerce: A systematic review and agenda for future research.
- Amazon Seller Central. (2025). <https://developer.amazonservices.com/>
- Batista, Prati, & Monard. (2004). A Study of the Behavior of Several Methods for Balancing machine Learning Training Data.
- Bobbitt, Z. (2021, May 25). *Augmented Dickey-Fuller Test in Python (With Example)*. Statology. Retrieved September 13, 2025, from <https://www.statology.org/dickey-fuller-test-python/>
- Chen, L., Mislove, A., & Wilson, C. (2016). An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace.
- Chopra, S., & Meindl, P. (2016). *Supply Chain Management: Strategy, Planning, and Operation*. Pearson.
- Evans, H. (2023, September 26). *Business Intelligence Architecture: Craft a Robust BI System*. Velvetech, LLC. Retrieved 2025, from <https://www.velvetech.com/blog/business-intelligence-architecture/>
- Géron, A. (2022). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly.
- He, H., & Ma, Y. (Eds.). (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley.
- I.G.L.M. Electronic srl. (n.d.).
- Kohavi, R. (2001). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.
- Liu, Kalaitzi, Wang, & Papalagnou. (2025). A Machine Learning Approach to Inventory Stockout Prediction.
- McAfee, Brynjolfsson, E., Davenport, T. H. D., Patil, & Barton. (2012). Big data: The management revolution. *Harvard Business Review*.

Seablod, Skipper, & Perktold. (2010, October 3). statsmodels: Econometric and statistical modeling with python." <https://www.statsmodels.org/stable/index.html>

Toogood, M. (2024, October 18). *Amazon FBA vs FBM: Which is right for you?* Sell on Amazon. Retrieved September 13, 2025, from <https://sell.amazon.com/blog/fba-vs-fbm>

Tuli, Mohammed, & Sachani. (2024). Supply Chain Optimization: Machine Learning Applications in Inventory Management for E-Commerce.

Zhao. (2024). Research on E-Commerce Retail Demand Forecasting Based on SARIMA Model and K-means Clustering Algorithm.