



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di laurea magistrale in Ingegneria Meccanica

Tesi di laurea

**Sviluppo di tools di intelligenza artificiale per la
programmazione della produzione**

**Development of artificial intelligence tools for
production scheduling**

Relatore: **Prof. Filippo Emanuele Ciarapica**

Tesi di laurea di: **Antonio Di Foglio**

A.A. 2021-2022

Indice

Capitolo 1

Che cos'è l'AI?	4
Utilizzo dell'AI nei giorni nostri.....	8
Tecniche di AI utilizzate per la programmazione della produzione.....	11
1.4 Probabilità di guasto	

Capitolo 2

Regole di associazione.....	16
Uso delle regole di associazione.....	17

Capitolo 3

Algoritmo Apriori.....	25
Algoritmo di crescita del pattern frequente (FP).....	28
Algoritmo ECLAT.....	30
Differenze tra Apriori e FP.....	30
Algoritmi nella letteratura.....	31

Capitolo 4

Introduzione e obiettivo.....	39
Librerie che implementato algoritmi di associazione.....	40
Sviluppo del tool.....	43
Conclusione.....	52

Bibliografia	55
---------------------------	----

Introduzione

L'intelligenza artificiale (IA) è una branca dell'informatica che si concentra sullo sviluppo di algoritmi e tecniche che permettono a un computer di svolgere compiti che normalmente richiederebbero capacità umane, come la percezione visiva, la comprensione del linguaggio e il prendere decisioni.

Le regole di associazione sono un tipo di algoritmo di intelligenza artificiale utilizzato per identificare relazioni tra elementi all'interno di un insieme di dati.

Questi algoritmi vengono spesso utilizzati nell'analisi dei dati di mercato per identificare le tendenze e le opportunità commerciali.

In questo lavoro viene presentato lo sviluppo di un tool di intelligenza artificiale che attraverso l'estrazione delle regole di associazione, aiuta l'azienda Matrix SRL nella programmazione della produzione.

Capitolo 1

Che cos'è l'AI?

L'Intelligenza Artificiale o Artificial Intelligence (AI) è un tema storicamente e scientificamente ricchissimo, che si rifà ad un'intima ispirazione dell'uomo: quella di creare una macchina in cui si riflettono appieno le proprie capacità. Gli ambiti applicativi sono innumerevoli e molti di questi potrebbero avere impatti importanti sulle attività di imprese e pubbliche amministrazioni, oltre che migliorare la vita delle persone. E non mancano le implicazioni etiche e filosofiche. L'idea di fondo dell'Intelligenza Artificiale è molto semplice: sviluppare delle macchine dotate di capacità autonome di apprendimento e adattamento che siano ispirate ai modelli di apprendimento umani. L'AI al giorno d'oggi può essere pensata come un'intersezione tra tecnologie che ragionano, interagiscono e imparano:

Ragionamento: Il ragionamento consente alle tecnologie AI di estrarre informazioni critiche da ampi set di dati strutturati e non strutturati, eseguire analisi di clustering e utilizzare l'inferenza statistica. Il livello di analisi a cui si è arrivati oggi è sempre più vicino alle capacità umane.

Interazione: L'interazione consente alle tecnologie AI di utilizzare metodi come la visione artificiale per vedere, la conversational AI e la linguistica computazionale per comunicare, imitando sempre di più l'essere umano, avvicinandosi da un punto di vista comportamentale

Apprendimento: Ciò che distingue realmente l'intelligenza artificiale dall'automazione intelligente però è la capacità della tecnologia di apprendere. L'apprendimento serve a diventare più intelligente nel tempo, così da poter replicare gesti e mansioni ormai appresi appieno dall'intelligenza. Solo l'AI possiede questa terza dimensione. Uno dei motivi per cui l'intelligenza artificiale è così promettente è

perché cambia il paradigma nel modo in cui abbiamo scritto il codice del software. Invece di programmare in ogni condizione "if, then ... except", dicendo a un motore di elaborazione cosa fare e come farlo, l'AI permette di risolvere dei compiti senza dover scrivere esplicitamente del codice specifico che spieghi come risolvere quel compito e quindi il motore può imparare e capire autonomamente come risolvere i vari compiti. In tal modo, l'AI affronta i problemi che la programmazione software tradizionale non ha mai potuto affrontare.

Ad oggi ci sono tre tipi di apprendimento diversi all'interno dell'AI:

1. **Supervised Learning (Apprendimento assistito):** Questa è la forma di apprendimento attualmente più comune; al sistema vengono forniti degli esempi campione riguardanti sia gli input che gli output e a partire da questi esegue una serie di tecniche basate sulla statistica (ma non solo) utili ad estrapolare la logica finale che collega input e output. Una volta che il sistema ha definito la logica finale, viene considerato "addestrato" e può applicare la funzione a qualsiasi nuova serie di informazioni. Per fare un esempio, questo tipo di apprendimento automatico può aiutare a prevedere la probabilità che ha un'azienda di fallire. Fornendo al sistema una grande quantità di dati sulle società (input) indicando se sono fallite o no (output), infatti, la macchina può estrapolare le tendenze sottostanti. Grazie a ciò il sistema può distinguere i segnali di avvertimento e quindi segnalare per tempo ad un'azienda la propria situazione di rischio o meno.

2. **Unsupervised Learning (Apprendimento non assistito):** L'apprendimento non assistito è quando una macchina viene presentata con un insieme di documenti o dati e poi calcola le cose da sola (gli esempi sono costituiti solo di input e non di output). Ad esempio, se una società desidera classificare una serie di contratti, la macchina potrebbe leggere ciascun documento e, in base al contesto, separarli automaticamente in categorie, ad esempio clausole come quella sulla proprietà intellettuale, la clausola sulla limitazione della responsabilità, la clausola di

indennizzo, ecc. Quindi, la macchina è in grado di definire un'ontologia rudimentale senza input umani.

3. **Reinforcement Learning (Apprendimento rafforzativo):** Conosciuto anche come apprendimento "orientato all'obiettivo", l'apprendimento di rinforzo è quando una macchina viene presentata con un obiettivo prefissato e poi lasciata libera per fare ciò che deve, pur rispettando i vincoli del contesto - compresi gli aggiustamenti e le rilavorazioni - fino a trovare i modi per raggiungere più efficacemente quell'obiettivo. Gli esempi più popolari di apprendimento di rinforzo si trovano nei videogiochi, in questi casi l'obiettivo della macchina è vincere il gioco. Si può vedere come può essere applicato anche nella pianificazione della produzione imparando a gestire una moltitudine di complessità per supportare i pianificatori nel raggiungimento degli obiettivi aziendali come livello di servizio, magazzini ed efficienza.

Il concetto di Artificial Intelligence è composta da due differenti tipologie di AI:

- **Intelligenza Artificiale Forte**
- **Intelligenza Artificiale Debole**

Nella filosofia dell'intelligenza artificiale, l'intelligenza artificiale forte è un concetto secondo il quale le opportune forme di AI possano ragionare e risolvere problemi, in sostituzione all'uomo; l'intelligenza artificiale forte sostiene che è possibile per le macchine diventare sapienti o coscienti di sé, ma non necessariamente devono mostrare processi di pensiero simili a quelli umani. Il termine intelligenza artificiale forte (AI forte, in inglese strong AI) fu originalmente coniato da John Searle. Searle affermava che «Secondo l'intelligenza artificiale forte, il computer non sarebbe soltanto, nello studio della mente, uno strumento; piuttosto, un computer programmato opportunamente è davvero una mente». Una teoria sviluppata dal filosofo Thomas Hobbes, diceva che l'intelligenza, ovvero la facoltà di ragionare, non è altro che un insieme di calcoli della mente umana. Alla base dell'Intelligenza

Artificiale forte troviamo la tecnologia dei sistemi esperti, cioè una serie di programmi che vogliono riprodurre, attraverso una macchina, le prestazioni e le conoscenze delle persone esperte in un determinato campo. Il sistema esperto opera attraverso 3 step differenti; il primo riguarda la base di partenza, che consiste nell'eseguire deduzioni e derivazioni logiche in base a una serie di assunzioni iniziali e di regole da applicare al problema. Il motore inferenziale è il secondo elemento e consiste nell'applicazione, in una determinata situazione, dei dati e delle nozioni, esso è costituito da tre componenti, i meta-algoritmi di ricerca associativa, i meta-algoritmi di ricerca comparativa e i meta-algoritmi di apprendimento. I meta-algoritmi di ricerca associativa estraggono gli elementi utili alla costruzione dell'algoritmo risolutivo del problema e forniscono la risposta alla domanda su quale conoscenza è necessaria per risolvere il problema. I meta-algoritmi di ricerca comparativa permettono di trovare la giusta sequenza e la modalità di applicazione delle informazioni estratte dagli algoritmi di ricerca associativa, fornendo la risposta alla domanda su come utilizzare la conoscenza a disposizione per risolvere il problema. I meta-algoritmi di apprendimento, infine, modificano il funzionamento, dei meta-algoritmi di ricerca associativa e di ricerca comparativa sulla base dell'esperienza acquisita consentendo l'apprendimento del sistema esperto durante il suo utilizzo, cosicché le informazioni acquisite per la soluzione di un problema accrescono la conoscenza del sistema e sono riutilizzate in futuro. L'ultimo modulo di un sistema esperto è l'interfaccia con l'utente che permette all'utente stesso di sottoporre il problema al sistema esperto e di ricevere la soluzione. Se la conoscenza di un esperto in un campo qualsiasi, può essere riversata in un calcolatore, questa macchina può sostituire con successo la persona? Gli studiosi che hanno creduto di più in questa branca dell'Intelligenza Artificiale, ovvero i funzionalisti, sostengono quindi che è possibile creare una macchina artificiale che sia, in tutto e per tutto, uguale e a volte anche superiore alla mente umana e per stabilire se un sistema artificiale sia o meno intelligente si effettua il Test di Turing, criterio suggerito dal matematico e filosofo britannico Alan Turing nell'articolo "Computing machinery

and intelligence” apparso nel 1950 sulla rivista Mind. Il Test di Turing prevede che, un uomo rinchiuso in una stanza, ponga delle domande ad un calcolatore attraverso una tastiera remota; se l’uomo non riesce a capire se dall’altra parte della stanza le risposte gli vengono fornite da un essere umano o da una macchina, allora siamo in presenza di un calcolatore intelligente. Tuttora però nessuna macchina ha mai superato con successo tale test, i fervidi sostenitori dell’Intelligenza Artificiale forte affermano che è semplicemente una questione di tempo.

In contrasto con l’intelligenza artificiale forte abbiamo l’intelligenza artificiale debole, essa si riferisce all'uso di programmi per studiare o risolvere specifici problemi o ragionamenti che non possono essere compresi pienamente o sono completamente al di fuori dei limiti delle capacità cognitive umane. Diversamente dall'intelligenza artificiale forte, quella debole non realizza un'auto-consapevolezza della macchina e non presenta l’intervallo tra vari livelli di abilità cognitive proprio dell'uomo, ma è esclusivamente un problem-solver (risolutore di problemi) specifico e, parzialmente, intelligente.

Utilizzo dell’AI nei giorni nostri

Ci sono tante applicazioni che utilizzano l’intelligenza artificiale, senza che ce ne accorgiamo. Viene usata per lo shopping in rete, le pubblicità, ricerche online, per le traduzioni automatiche, nei veicoli e negli ultimi anni è servita nella lotta al COVID 19, un utilizzo banale è il controllo della temperatura nei luoghi pubblici. L’avvento del COVID 19 ha portato ad una nuova frontiera futura per l’AI come viene detto in “Long-COVID diagnosis: From diagnostic to advanced AI-driven models”. Lo scopo di questo manoscritto è quello di esaminare diversi aspetti della sindrome long-COVID-19 dalla presentazione clinica alla diagnosi, evidenziando il notevole impatto che l'IA può avere. Ad esempio, è stato dimostrato che l'AI è in grado di differenziare la polmonite COVID-19 dalla polmonite comune e da altre patologie polmonari “Using Artificial Intelligence to Detect COVID-19 and Community-acquired

Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy”. Nello scritto “, The characteristics and evolution of pulmonary fibrosis in COVID-19 patients as assessed by AI-assisted chest HRCT, PLoS ONE” gli autori hanno riportato che tramite l’AI il punteggio dell’infiammazione ha mostrato una buona correlazione con il punteggio quantitativo della fibrosi polmonare, concludendo che la tecnologia AI-assistita della TAC del torace può fornire dati qualitativi e quantitativi per analizzare l’evoluzione a lungo termine della fibrosi polmonare. Con l’avvento di internet e dei social media l’individuazione di notizie false e fuorvianti è diventata una priorità assoluta per ricercatori e professionisti. Nonostante i numerosi sforzi in quest’area, molte domande rimangono senza risposta sulla progettazione ideale degli interventi, in modo da informare efficacemente i consumatori di notizie. Possono essere forniti spunti, affinché l’AI possa contribuire a mitigare le conseguenze negative della disinformazione come viene discusso in “Tailoring heuristics and timing AI interventions for supporting news veracity assessments”. Viene detto che per individuare le fake news, si è lavorato molto sullo sviluppo di metodi di IA con l’obiettivo di supportare le valutazioni dei consumatori di notizie sulla veridicità degli articoli. Tuttavia, c’è relativamente poco lavoro sul fatto che questi metodi siano effettivamente utili nei sistemi rivolti all’utente finale. Il successo, anche se i consigli dell’algoritmo sono molto accurati, non è semplice. The Bipartisan Press è un publisher che ha creato un algoritmo di AI che utilizza il machine learning per analizzare e “imparare” com’è fatta una notizia con opinioni di destra e una di sinistra. L’azienda ha allenato il loro algoritmo con un corposo set di dati. Ha fatto analizzare la struttura di centinaia di articoli giornalistici e ora, a loro detta, il sistema è in grado di valutare il tipo e il livello di faziosità politica di una notizia con una precisione del 96%. D’altro canto, l’AI, può essere utilizzata per creare bot generatori di notizie false, ma l’obiettivo del The Bipartisan Press è quello fornire algoritmi e altri strumenti che ci aiutano a valutare fonti e informazione per leggere meglio e capire di più l’attualità. Google fornisce agli utenti una tecnologia chiamata NewsGuard, essa permette di riconoscere le fonti di informazione e se sono affidabili

o meno. NewsGuard valuta i siti e i publisher in base ai loro dati storici, ovvero le pubblicazioni passate. Se un sito ha pubblicato notizie tendenziose o fake news, questo viene registrato nel database e il punteggio di affidabilità del publisher si abbassa. Questi algoritmi dimostrano che le nuove tecnologie possono essere utilizzate per creare strumenti utili e benefici, e non solo nuove minacce per la democrazia. Anche nel settore pubblico l'interesse per l'implementazione di software che sfruttano l'intelligenza artificiale è in crescita. In singoli servizi pubblici sono già state effettuate le prime implementazioni e ricerche; tuttavia, nel settore pubblico manca una migliore comprensione dell'accettazione di questa tecnologia da parte dei cittadini, in quanto le conoscenze acquisite nel settore privato non possono essere trasferite direttamente. Nell'articolo "Artificial intelligence in public services: When and why citizens accept its usage" vengono condotti esperimenti di policy-capturing per analizzare l'accettazione dell'AI in alcuni scenari rappresentativi. Sulla base della teoria del ragionamento comportamentale, sono state raccolte prove da 329 partecipanti. I risultati hanno mostrato che le soluzioni di AI nei servizi pubblici generali sono preferite a quelle fornite dagli esseri umani, ma i servizi specifici sono ancora di competenza umana. Ulteriori analisi mostrano che i principali fattori di accettazione sono le ragioni contro l'IA e i risultati contribuiscono a comprendere quando e perché l'IA viene accettata nei servizi pubblici. Per portare avanti gli esperimenti nei vari scenari, per capire il grado di accettazione dell'AI nel settore pubblico, sono stati raccolti dati nei mesi di aprile e maggio 2020 utilizzando Clickworker, è stata scelta l'Australia come scenario di ricerca, in quanto il Paese è digitalmente avanzato nel settore pubblico e privato. I risultati mostrano che l'accettazione dell'IA è significativamente inferiore a quella degli esseri umani che forniscono lo stesso servizio specifico, ma per gli ambiti tematici generali mostrano che l'IA è sempre più accettata rispetto allo stesso servizio effettuato dagli umani, anche se in alcuni casi la differenza non è significativa. L'Intelligenza Artificiale è utilizzata anche nei Social e nei motori di ricerca. Social quali Facebook, Instagram e TikTok e motori di ricerca come Youtube e Google. Nei Social Network viene

utilizzata con i post consigliati, l'intelligenza attraverso una serie di parametri ti propone dei contenuti che potrebbero piacerti in base a quelli che hai già apprezzato sulla piattaforma. Essa è quindi in grado di capire quali sono i contenuti simili attraverso una sequenza di dati e informazioni che analizza. Nei Motori di Ricerca è invece in grado di capire cosa stai cercando in base a cosa hai cercato prima.

Se cerchi per esempio "radice" e in precedenza hai consultato siti di matematica, il motore ti consiglierà risultati riguardanti la radice matematica e non la radice di una pianta.

Tecniche di AI utilizzate per la programmazione della produzione

Numerose aziende vogliono implementare l'intelligenza artificiale al proprio interno, in modo da rendere il proprio reparto produttivo più intelligente, più veloce, più competitivo e redditizio. Troviamo conferma in uno studio svolto dal Fortune Knowledge Group che rivela *"l'82% dei dirigenti ha intenzione di implementare progetti di intelligenza artificiale all'interno delle proprie aziende nei prossimi tre anni"*. In un impianto produttivo gli eventi stocastici, come gli ordini urgenti, gli eventi di stock-out e i guasti locali, hanno un impatto importante sulle prestazioni della produzione distribuita, ma è difficile prevederli e tenerne conto nella programmazione delle attività produttive. L'intelligenza artificiale può fornire una conoscenza in modo da sincronizzare efficacemente gli eventi, così da avere vantaggi misurabili come la riduzione dei ritardi di comunicazione e, di conseguenza, una migliore reattività del sistema ai cambiamenti di produzione e alle nuove esigenze di schedulazione che si presentano. Nell'articolo "Improving the remote scheduling of distributed production with process statistics and AI techniques" vengono discusse le implementazioni di queste nuove tecnologie e le loro applicazioni nel caso industriale. Per avere una programmazione efficace si possono collegare i modelli di

simulazione del sistema fisico e i moduli di pianificazione; i modelli di simulazione forniscono stime sullo stato degli impianti di produzione nel momento in cui il lotto successivo deve essere rilasciato per la produzione e sulla base di queste stime e del piano di produzione corrente, i moduli di schedulazione distribuiscono i lavori di produzione pertinenti ai diversi impianti di produzione. Questo approccio ha dimostrato che introducendo eventi di sincronizzazione specifici del processo, il ritardo medio di comunicazione può essere ridotto del 25-30%. Un ulteriore lavoro ha anche dimostrato quantitativamente come le prestazioni di schedulazione possano beneficiare di questo miglioramento a livello di comunicazione. In particolare, è stato dimostrato che una riduzione dei ritardi di comunicazione riduce significativamente il numero medio di scadenze di produzione non rispettate (ad esempio, consegne di prodotti in ritardo) e il ritardo cumulativo di quelle non rispettate. La ricerca descritta in questo articolo ha affrontato l'impatto di eventi critici, come i guasti dei componenti, sulla programmazione a distanza di processi produttivi coordinati e sono state fornite prove che dimostrano che le prestazioni di uno strumento di simulazione e schedulazione distribuita può essere migliorato in modo significativo se si utilizzano statistiche di processo adeguate per personalizzare la tempistica degli eventi di sincronizzazione. È stata poi ideato un secondo modello di ricerca e sono state fornite prove per dimostrare che anticipando l'impatto dei guasti locali sulla produttività degli impianti coordinati, sotto forma di guasti effettivi, è possibile mantenere nel tempo elevati livelli di utilizzo delle risorse e di produttività in tutto il sistema produttivo. Per questo scopo è stato ipotizzato uno strumento di intelligenza artificiale che predice i guasti modulari. Esso è basato su reti neurali artificiali in grado di fornire stime rapide di questi impatti e di consentire la riprogrammazione locale delle attività in funzione dei guasti.

Leggendo l'articolo "Multi-layered coding-based study on optimization algorithms for automobile production logistics scheduling" ci accorgiamo che con l'accelerazione della globalizzazione economica, la concorrenza tra le industrie manifatturiere è diventata sempre più agguerrita. Nell'ambito automobilistico il

problema della programmazione logistica della produzione di automobili influisce sulla capacità delle imprese di competere, sviluppando modelli matematici si possono risolvere questo tipo di problemi e questo documento discute l'uso di tecnologie dirompenti, come AI, IoT, Big data, ecc. Viene fornita una suddivisione nell'uso dell'intelligenza artificiale nella logistica: da una parte l'AI può sostituire parte della manodopera con dispositivi intelligenti, come camion senza equipaggio, AMR, veicoli di distribuzione senza equipaggio, droni, robot per il servizio clienti, ecc. In secondo luogo migliora l'efficienza delle risorse umane attraverso la computer vision, l'apprendimento automatico, l'ottimizzazione delle operazioni e altre tecnologie o sistemi software basati su algoritmi, come i sistemi di gestione del magazzino, sistemi di programmazione delle attrezzature e sistemi di distribuzione degli ordini. Un esempio può essere Amazon, il gigante mondiale dell'e-commerce, possiede 100.000 robot Kiva, che gestiscono ordini, magazzino e altro. Il robot Kiva è una tecnologia intelligente per la logistica, è l'applicazione perfetta dell'AI per la trasformazione del settore logistico. Kiva è dotato di un sistema multi-intelligente e può effettuare operazioni come studiare, coordinare, prendere decisioni che coinvolgono varie discipline e programmare computazionalmente processi decisionali incerti, pianificare il data mining, eseguire l'ottimizzazione classica e altre funzioni. Ci sono anche modelli per ottimizzare i programmi di manutenzione e produzione in modo predittivo e automatico, come viene detto in "A predictive maintenance model for optimizing production schedule using deep neural networks", qui il modello è chiamato Predictive Maintenance (PdM) & Schedule (PdMS). La valutazione del modello è stata eseguita sulla base di un set di dati preparato per l'apprendimento di classificazione supervisionata. Il set di dati conteneva misure di tensione, rotazione, pressione e vibrazioni telemetriche ogni ora (ciclo) per un anno. Insieme alla proposta del modello, sono stati convalidati, creati e combinati gli indici di degrado. In questo articolo, dove vediamo lo sviluppo del modello PdMS, notiamo che integra autonomamente i programmi di manutenzione e produzione. Inoltre sono stati utilizzati modelli ML insieme agli algoritmi per i problemi di schedulazione e la

creazione del modello ha comportato la sfida di identificare modelli e strategie differenziate senza perdere la possibilità di un uso generico.

Una delle ultime tendenze della ricerca manifatturiera dell'UE è quella di aumentare la riconfigurabilità dei sistemi di produzione introducendo lavoratori mobili a doppio braccio in grado di collaborare con gli esseri umani in ambienti dinamici e mutevoli. In “AI based combined scheduling and motion planning in flexible robotic assembly lines” si discute di questo tema presentando un quadro decisionale a più livelli che si concentra sul bilanciamento dinamico del lavoro tra gli operatori umani e le risorse robotiche, proponendo tecniche di intelligenza artificiale per generare sequenze di compiti efficienti, garantendo al contempo la generazione di percorsi ottimali e senza collisioni per le risorse assegnate, testando gli sviluppi in un caso di studio dell'industria automobilistica. L'effetto sinergico di operatori umani che lavorano con robot flessibili è stato identificato come uno dei principali fattori abilitanti per aumentare la reattività dei sistemi di produzione. Negli ultimi anni, sono stati pubblicati diversi approcci su come affrontare la programmazione dei compiti tra squadre di robot umani che mirano a una cooperazione armonica ed efficiente, considerando non solamente gli aspetti di controllo robotico di basso livello, come la reazione di sicurezza, il movimento accettabile o la manipolazione consapevole dell'uomo.

Il presente articolo si propone di introdurre un framework che, sfruttando le tecniche di Intelligenza Artificiale (IA) cerca di distribuire in modo efficace il lavoro tra le risorse umane e mobili e di ricavare piani di movimento ottimizzati per i lavoratori robot. Per consentire il bilanciamento dinamico del carico di lavoro tra le risorse umane e robotiche disponibili, è stato proposto un approccio decisionale a più livelli. A livello di linea su come distribuire le operazioni di assemblaggio, e a livello di risorse su come eseguire in modo ottimale i compiti assegnati, minimizzando percorsi privi di collisioni. Per concludere è stata proposta un'architettura aperta nel senso che si basa sul framework modulare ROS che è open e che supporta l'interfacciamento

con diversi tipi di robot e a diverse marche di robot. In questo modo, è facile integrare nuovi tipi di risorse nel processo decisionale. Allo stesso tempo, l'approccio unificato alla modellazione del mondo digitale permette di interpretare diverse sequenze di assemblaggio e quindi di supportare la programmazione delle operazioni per l'assemblaggio di diversi componenti.

Capitolo 2

Regole di associazione

Nel data mining, le regole di associazione sono uno dei metodi per estrarre relazioni nascoste tra i dati. Furono introdotte le regole di associazione per la scoperta di regolarità all'interno delle transazioni registrate nelle vendite dei supermercati, ad esempio, la regola [cipolle, patate] \square [hamburger] individuata nell'analisi degli scontrini di un supermercato indica che se il cliente compra insieme cipolle e patate è probabile che acquisti anche della carne per hamburger. Tale informazione può essere utilizzata come base per le decisioni riguardanti le attività di marketing, come ad esempio le offerte promozionali o il posizionamento dei prodotti negli scaffali. Le regole di associazione sono anche usate in molte altre aree, quali il Web mining, la scoperta di anomalie e la bioinformatica.

La definizione originale di Regola di Associazione, attribuita a Rakesh Agrawal, informatico che fino a poco tempo fa era Technical Fellow presso i Microsoft Search Labs, è rappresentata come segue.

Considerando l'insieme I di n attributi binari, oggetti o item, si definisce l'insieme $I = [i_1, i_2, i_3, \dots, i_n]$ e l'insieme di transazioni o dataset $D = [t_1, t_2, t_3, \dots, t_n]$. Ogni transazione che appartiene all'insieme D è in possesso di un codice identificativo (ID) e contiene un sottoinsieme degli n attributi contenenti nell'insieme I . Una regola di associazione è definita come un'implicazione nella forma $X \rightarrow Y$ dove $(X, Y) \in I$ e $X \cap Y = \emptyset$. L'insieme di oggetti X e Y vengono chiamati rispettivamente antecedente e conseguente della regola. Per spiegare meglio questo concetto

possiamo avvalerci di un esempio molto banale. In un supermercato possiamo avere un insieme di oggetti formato da $I = [\text{pane, burro, latte, hamburger, birra, ...}]$

ID	Latte	Pane	burro	Hamburger	Birra
1	1	1	0	0	0
2	0	0	1	1	0
3	0	0	0	0	1
4	1	1	1	0	0
5	0	1	0	0	0

Il database contenente gli oggetti è rappresentato nella tabella a destra, dove 1 indica la presenza di un oggetto in una transazione e 0 l'assenza. Un esempio di regola potrebbe essere $[\text{burro, pane}] \rightarrow [\text{latte}]$ ed indica che se un cliente acquisterà burro e pane, comprerà allo stesso momento anche il latte.

Nelle regole di associazione troviamo molti elementi, tra cui il database transazionale, ossia un archivio nel quale ogni istanza corrisponde ad una transazione. Nel nostro caso il database transazionale è rappresentato dalla tabella qui sopra, e le transazioni sono identificate dai numeri che vanno da 1 a 5 sotto la colonna ID. Una transazione è formata da un elenco di oggetti, chiamati anche item, che vanno a comporre la transazione; nel nostro caso la transazione numero 4 è composta dagli item latte, pane e burro. I database transazionali non sono così semplici ma, possono essere visti come database relazionali, con la possibilità di attributi multi-valore, per esempio ad ogni item può essere attribuito un valore. Ricapitolando una regola associativa collega tra di loro gli attributi di un insieme di dati, ed ha la forma “Corpo \rightarrow Testa [supporto, confidenza]” dove il corpo e la testa sono itemset, ossia congiunzione di item, e il supporto e la confidenza sono valori percentuali che indicano la valenza e la forza della regola di associazione. Di solito per supporto e

confidenza si fissano dei valori di soglia minima e si chiamano regole associative forti quelle regole per le quali supporto e confidenza superano i valori di soglia.

Le regole associative possono essere classificate:

-regole associative booleane e quantitative

> booleana se in tutti gli item $\text{attr}(x,v)$, v è un singolo valore, dove x indica la generica istanza del nostro insieme di dati, attr è un attributo che compone il nostro insieme di istanze e v è uno dei possibili valori (o un insieme di valori) assunti da attr

> quantitativa se coinvolge attributi numerici e v è un insieme di valori, nella maggior parte dei casi un intervallo.

-regole associative mono-dimensionali e multi-dimensionali: a seconda del numero di attributi diversi coinvolti.

Uso delle regole di associazione

In letteratura troviamo moltissimi articoli che parlano delle regole di associazione e del loro utilizzo in svariati campi si passa dall'uso di queste regole in campo industriale, o nella medicina dove sono state molto utili in questo periodo per trovare la connessione tra i vari sintomi del COVID-19 e la malattia stessa. Vengono usate queste regole insieme ai loro algoritmi ,anche per facilitare la vita del cittadino comune, nei servizi pubblici e in molteplici altri campi.

Però, come troviamo scritto in “FR-Tree: A novel rare association rule for big data problem”, in alcune situazioni, trovare una regola di associazione rara è più importante dell'insieme di elementi frequenti, perché le regole rare rappresentano casi, attività o eventi rari nelle applicazioni del mondo reale descritte in precedenza. In questo articolo, notiamo che è essenziale estrarre attività critiche eccezionali da vasti dati di routine e viene proposto l'algoritmo chiamato FR-Tree per estrarre le

regole di associazione e produrre regole essenziali. Con questo lavoro, gli autori, mirano a dimostrare che questo algoritmo è adatto a estrarre regole di associazione rare con un'elevata confidenza e in più genera, filtra e classifica le regole essenziali, frequenti o rare. Da considerare è il fatto che le regole rare sono state prodotte senza la necessità di impostare una soglia aggiuntiva, pertanto, l'algoritmo proposto nel testo ha un vantaggio incomparabile con le altre tecniche di regole di associazione rare. Infine le regole generate sono state testate utilizzando set di dati noti e le prestazioni sono state confrontate con le altre tecniche di regole di associazione rare, ed i risultati hanno dimostrato che il metodo proposto ha superato le tecniche di regole di associazione rare esistenti.

Negli ultimi anni la parola più usata nel mondo è stata senza dubbio COVID-19, la troviamo dappertutto, giornali, tv e soprattutto social. E' molto interessante l'articolo "Mining association rules from COVID-19 related twitter data to discover word patterns, topics and inferences" che usa la pandemia COVID-19 come caso d'uso e analizza i tweet raccolti tra febbraio e agosto 2020, per estrarre regole di associazione e conoscenze sull'atteggiamento del pubblico nei confronti delle crisi mondiali. E' stata proposta una metodologia che comprende l'estrazione di argomenti e tecniche di visualizzazione, come le WordCloud, per formare cluster o temi di opinioni, vengono poi utilizzate le Association Rule Mining (ARM) per scoprire insiemi di parole frequenti e generare regole che deducono gli atteggiamenti degli utenti. L'obiettivo che si sono posti è stato quello di utilizzare ARM come tecnica di post-elaborazione per migliorare il risultato di qualsiasi metodo di estrazione di argomenti e per questo, solo gli insiemi di parole forti sono state memorizzate dopo aver scartato quelle banali. Il lavoro ha dimostrato che la metodologia presentata nell'articolo ha facilitato la produzione di risultati più accurati e generalizzabili, esponendo al contempo le implicazioni relative agli atteggiamenti degli utenti dei social media.

Un tema molto discusso è quello relativo all'uso dei social e delle varie personalità influenti, leggendo "Graph-based modeling using association rule mining to detect

influential users in social networks” ci rendiamo conto che la diffusione delle informazioni è un campo di ricerca importante e interessante nell'ambito dell'analisi delle reti sociali ed è al centro di molti studi e applicazioni di estrazione e previsione della conoscenza. La maggior parte di questi studi, nonostante si è concentrata sull'identificazione degli utenti più influenti e sulla previsione della partecipazione degli utenti, ha ancora bisogno di approcci basati sull'estrazione di regole di associazione e sulla teoria dei grafi. In questo studio, è stato dato un contributo alla ricerca in questo campo introducendo un nuovo approccio basato sui grafi che applica l'estrazione di regole di associazione per individuare gli utenti influenti. E' stato sostenuto che gli utenti si influenzano a vicenda e che è possibile prevedere gli interessi e la partecipazione di un utente in base alle interazioni precedenti nella rete sociale. Sono stati introdotti nuovi concetti e algoritmi per una caratterizzazione più efficiente degli utenti influenti ed è stato sviluppato un approccio efficace per la scoperta degli influencer utilizzando tecniche di regole di associazione per estrarre le relazioni nascoste tra gli utenti. Gli autori, per illustrare l'efficienza del loro approccio, hanno condotto esperimenti su 25 grafi di diffusione generati e i risultati hanno mostrato che il loro approccio può raggiungere un alto livello di prestazioni in termini di tempo di calcolo per reti di grandi dimensioni.

In molti articoli troviamo l'uso delle regole di associazione per verificare le cause di possibili incidenti sui luoghi di lavoro, le correlazioni tra le azioni commesse dai lavoratori e le cause degli incidenti. Si discute di come le azioni non sicure dei lavoratori marittimi sono state ampiamente riconosciute come le cause dirette degli incidenti in mare, nell'articolo “On the causation of seafarers' unsafe acts using grounded theory and association rule”. Nonostante gli studi precedenti hanno dimostrato il ruolo significativo degli atti insicuri e hanno fornito numerosi fattori che influenzano gli atti insicuri dei lavoratori marittimi, non era stata ancora esplorata la complessa relazione tra i fattori.

In questo studio, gli autori, hanno usato la teoria degli insiemi grezzi integrandola con l'estrazione di regole di associazione per esplorare la potenziale causalità degli atti di insicurezza dei lavoratori marittimi. I risultati hanno evidenziato lo sviluppo di un quadro di analisi teorica degli atti di insicurezza dei lavoratori e la causalità degli atti di insicurezza, che viene presentata da tre prospettive: "fattori→ atti non sicuri", "auto associazione dei fattori" e "auto associazione degli atti non sicuri", fornendo anche le contromisure di sicurezza.

Troviamo trattato l'argomento degli incidenti marittimi anche nel testo "A hybrid model for marine accident analysis based on Bayesian Network (BN) and Association Rule Mining (ARM)". Sono stati fatti studi su incidenti non denunciati al trasporto marittimo e sono stati presi dati relativi agli incidenti professionali non denunciati su pescherecci turchi di lunghezza totale pari o superiore a 12 metri, poi analizzati utilizzando i metodi della rete bayesiana (BN) e dell'Association Rule Mining (ARM). Utilizzando l'algoritmo Predictive Apriori sono state stabilite le regole per il verificarsi degli infortuni sul lavoro sui pescherecci, tenendo conto di variabili come le condizioni del giorno, la lunghezza, le condizioni del mare e il tipo di nave. Queste regole hanno permesso di capire come si sono verificati gli infortuni sul lavoro sui pescherecci, dunque hanno definito i requisiti minimi per il verificarsi di incidenti sulle imbarcazioni da pesca.

Al giorno d'oggi, una parte significativa delle merci e dei passeggeri viene trasportata su autostrade extraurbane con veicoli prevalentemente ad alta velocità, e di conseguenza, queste autostrade sono molto soggette a collisioni con diverse lesioni. Questo argomento viene trattato in "Classification and association rule mining of road collisions for analyzing the fatal severity, a case study". In questo scritto vengono utilizzati tre algoritmi, tra cui Artificial Neural Network (ANN), Support Vector Machine (SVM) e Random Forests (RF), per costruire modelli di classificazione per la gravità della fatalità di 2355 record di collisioni mortali avvenute nel periodo 2007-2009 sulle strade di 8 Stati degli USA con diversi fattori

legati al conducente, all'ambiente e alla strada. Per ogni classificatore sono state generate mappe di rischio previste e l'importanza dei fattori che vi hanno contribuito è stata valutata in base alla diminuzione media dell'accuratezza e alla diminuzione media dell'indice di Gini. Infine, è stata eseguita l'estrazione di regole di associazione con l'algoritmo Apriori per estrarre le regole di collisione. RF si è dimostrato il migliore, in quanto ha superato gli altri metodi in termini di accuratezza complessiva e tassi di kappa, rispettivamente del 94% e del 92%, e inoltre sono state estratte sessantotto regole di associazione con l'algoritmo Apriori.

Un articolo molto interessante è “Multidrug resistance analysis method for pathogens of cow mastitis based on weighted-association rule mining and similarity comparison” dove è viene proposto un algoritmo di estrazione di regole di associazione ponderate Apriori, per risolvere il problema dei numerosi e disordinati dati sulla resistenza ai farmaci, e della mancanza di correlazioni estese, negli allevamenti da latte. Viene analizzato il crescente problema della resistenza ai farmaci, in particolare la resistenza al multi farmaco (MDR), rappresenta una grande minaccia per il controllo della mastite, una delle malattie più comuni e causa le maggiori perdite economiche negli allevamenti da latte. Per comprendere le regole della MDR nei batteri della mastite bovina a livello nazionale, sono stati isolati e identificati i principali batteri delle vacche affette da mastite in allevamenti su larga scala in Cina e sono stati condotti test di sensibilità ai farmaci per stabilire un set di dati sulla resistenza ai farmaci. Gli autori hanno lavorato su un set di dati sulla resistenza ai farmaci che mostrava che gli stafilococchi sono i principali batteri isolati dalla mastite delle vacche da latte in Cina. Sulla base dell'algoritmo delle regole di associazione, sono state identificate le regole importanti tra le diverse resistenze agli antibiotici in questo set di dati. I risultati dimostrano che i batteri sono altamente resistenti alla penicillina, alla gentamicina e all'ampicillina e che la maggior parte degli altri antibiotici sono collegati a questi tre antibiotici. Infine, le correlazioni elevate e le regole principali nei diversi batteri sono state confermate da un metodo di confronto di similarità. Il modello di valutazione e le conclusioni di questo studio

sono potenzialmente utili per valutare l'evoluzione dei pattern MDR, fornendo una base scientifica alle autorità competenti per guidare l'uso razionale degli antibiotici nell'industria agricola.

Nel contesto dei big data, il giudizio scientifico sulla tendenza o sullo stato futuro degli atti non sicuri dei controllori del traffico aereo (ATC) svolge un ruolo importante nella prevenzione degli incidenti non sicuri, come possiamo leggere nell'articolo "Risk prediction and early warning for air traffic controllers' unsafe acts using association rule mining and random forest". Gli autori, sulla base dei precedenti studi di previsione e di allarme precoce, per questo studio hanno scelto i metodi dell'analisi delle correlazioni e della modellazione non lineare per prevedere e avvisare scientificamente gli atti non sicuri dei controllori del traffico aereo. In primo luogo, sono state estratte le regole di associazione tra i fattori di rischio e gli atti non sicuri dei controllori, e sulla base dell'estrazione di queste regole, il modello di previsione e di allarme precoce è stato creato con l'algoritmo della foresta casuale. I risultati hanno mostrato che il numero di regole di associazione bidimensionali è il più alto, seguito da regole di associazione monodimensionali, mentre le regole di associazione quadridimensionali sono le più basse. Per quanto riguarda la previsione degli errori e delle violazioni, la deviazione tra il valore previsto e quello effettivo è piccola, il che indica che l'accuratezza della previsione è elevata. Inoltre, l'applicazione delle regole di associazione e del modello della foresta casuale può prevedere efficacemente gli errori e le violazioni degli ATCers e inviare segnali ai dipartimenti competenti in base alla soglia e al livello di allarme per segnalare le situazioni di pericolo, riducendo così le perdite causate dalle azioni non sicure degli ATCers.

Le regole di associazione sono utilizzate molto in campo medico, come possiamo vedere nello studio proposto dagli autori di "Comorbidity combinations in schizophrenia inpatients and their associations with service utilization: A medical record-based analysis using association rule mining". Il lavoro è stato incentrato sulle

comorbilità che sono comuni tra i pazienti con schizofrenia, però la prevalenza delle combinazioni di comorbilità e le loro associazioni con l'utilizzo dei servizi di ricovero e la riammissione sono state scarsamente esplorate. Per portare avanti la ricerca, i dati sono stati estratti dalle schede di dimissione dei pazienti la cui diagnosi primaria era di disturbo dello spettro schizofrenico (ICD-10: F20-F29). Lo studio ha incluso i dati di 8252 pazienti, sono state identificate le 30 comorbilità più frequenti nelle diagnosi secondarie dei pazienti e poi si è utilizzato il metodo dell'association rule mining (ARM) per ricavare combinazioni di comorbilità associate alla durata della degenza (LOS), alla spesa giornaliera e alla riammissione a un anno. Possiamo dedurre che la maggior parte delle combinazioni di comorbilità identificate dall'ARM è stata significativamente associata a una LOS più lunga (≥ 70 giorni), poche sono state associate a spese giornaliere più elevate e poche a riammissioni.

Capitolo 3

ALGORITMI USATI PER REGOLE DI

ASSOCIAZIONE

Algoritmo APRIORI

Sono stati sviluppati diversi algoritmi statistici per implementare il mining di regole di associazione e Apriori è uno di questi algoritmi. Apriori tenta di operare su record di database, in particolare record transazionali o record che includono determinati numeri di campi o elementi. È uno dei numerosi algoritmi che utilizzano un "approccio dal basso verso l'alto" per contrastare in modo incrementale record complessi ed è utile nei complessi progetti di machine learning e intelligenza artificiale di oggi. Ci sono tre componenti principali dell'algoritmo Apriori e per spiegare questi tre componenti userò un esempio come aiuto: supponiamo di avere un record di 1.000 transazioni di clienti e di voler trovare il supporto, la fiducia e l'incremento per due articoli, ad esempio hamburger e ketchup. Su mille transazioni, 100 contengono ketchup mentre 150 contengono un hamburger. Su 150 transazioni in cui viene acquistato un hamburger, 50 transazioni contengono anche ketchup. Utilizzando questi dati, vogliamo trovare il supporto, la fiducia e l'incremento. Il supporto si riferisce alla popolarità predefinita di un articolo e può essere calcolato trovando il numero di transazioni contenenti un particolare articolo diviso per il numero totale di transazioni. Supponiamo di voler trovare supporto per l'elemento B. Questo può essere calcolato come:

$$\text{Supporto}(B) = (\text{Transazioni contenenti } (B)) / (\text{Totale transazioni})$$

Ad esempio, se su 1000 transazioni, 100 contengono Ketchup, il supporto per l'elemento Ketchup può essere calcolato come:

$$\text{Supporto(Ketchup)} = (\text{Transazioni contenenti Ketchup})/(\text{Totale Transazioni})$$

$$\text{Supporto(Ketchup)} = 100/1000 = 10\%$$

Dopo il supporto abbiamo la fiducia che si riferisce alla probabilità che un articolo B venga acquistato anche se viene acquistato l'articolo A. Può essere calcolato trovando il numero di transazioni in cui A e B vengono acquistati insieme, diviso per il numero totale di transazioni in cui A viene acquistato. Matematicamente, può essere rappresentato come:

$$\text{Confidenza}(A \rightarrow B) = (\text{Transazioni contenenti (sia A che B)})/(\text{Transazioni contenenti A})$$

In un ipotetico problema, possiamo avere 50 transazioni in cui hamburger e Ketchup sono stati acquistati insieme e altre 150 transazioni vengono acquistati hamburger. Quindi possiamo trovare la probabilità di acquistare ketchup quando viene acquistato un hamburger può essere rappresentato come fiducia di hamburger \rightarrow Ketchup e può essere matematicamente scritto come:

$$\text{Confidenza}(\text{Hamburger} \rightarrow \text{Ketchup}) = (\text{Transazioni contenenti (sia hamburger che Ketchup)})/(\text{Transazioni contenenti hamburger})$$

$$\text{Confidenza}(\text{hamburger} \rightarrow \text{Ketchup}) = 50/150 = 33.3\%$$

Successivamente abbiamo il sollevamento o lift che si riferisce all'aumento del rapporto di vendita di B al momento della vendita di A. Lift (A \rightarrow B) può essere calcolata dividendo Confidenza(A \rightarrow B) diviso per Supporto(B):

$$\text{Lift}(A \rightarrow B) = (\text{Confidenza}(A \rightarrow B))/(\text{Supporto}(B))$$

Facendo riferimento al caso precedente:

$$\text{Lift}(\text{hamburger} \rightarrow \text{Ketchup}) = (\text{Confidenza}(\text{hamburger} \rightarrow \text{Ketchup})) / (\text{Supporto}(\text{Ketchup}))$$

$$\text{Lift}(\text{hamburger} \rightarrow \text{Ketchup}) = 33.3/10 = 3.33$$

Lift fondamentale ci dice che la probabilità di acquistare un hamburger e un ketchup insieme è 3,33 volte superiore alla probabilità di acquistare semplicemente il ketchup. Un aumento di 1 significa che non vi è alcuna associazione tra i prodotti A e B. Un aumento di 1 significa che è più probabile che i prodotti A e B vengano acquistati insieme. Infine, Lift inferiore a 1 si riferisce al caso in cui è improbabile che due prodotti vengano acquistati insieme.

I passaggi coinvolti nell'implementazione dell'algoritmo possono essere molteplici, perché per insiemi di dati di grandi dimensioni, possono esserci centinaia di elementi in centinaia di migliaia di transazioni. L'algoritmo Apriori cerca di estrarre regole per ogni possibile combinazione di elementi. Ad esempio, l'incremento può essere calcolato per l'elemento 1 e l'elemento 2, l'elemento 1 e l'elemento 3 e così via. questo processo può essere estremamente lento a causa del numero di combinazioni. Per accelerare il processo, si possono eseguire alcuni passaggi descritti successivamente:

1) Molto utile è impostare un valore minimo per il supporto e la fiducia. Ciò significa che siamo interessati solo a trovare regole per gli elementi che hanno una certa esistenza di default (ad es. supporto) e hanno un valore minimo per la co-occorrenza con altri elementi (ad es. confidenza).

2) Successivamente si estraggono tutti i sottoinsiemi aventi un valore di supporto superiore alla soglia minima.

3) Una volta estratti i sottoinsiemi si selezionano tutte le regole dai sottoinsiemi con un valore di confidenza superiore alla soglia minima.

4) Infine si ordinano le regole in ordine crescente di Lift

Algoritmo di crescita del pattern frequente (FP)

Questo algoritmo, in molti dicono, è un miglioramento del metodo Apriori. Viene generato un pattern frequente senza la necessità della generazione di candidati. L'algoritmo di crescita FP rappresenta il database sotto forma di un albero chiamato albero pattern frequente o albero FP. Questa struttura ad albero manterrà l'associazione tra i set di elementi. Il database viene frammentato utilizzando un elemento frequente e questa parte frammentata verrà chiamata 'frammento di pattern'. Vengono analizzati i set di elementi di questi modelli frammentati. Pertanto, con questo metodo, la ricerca di set di elementi frequenti viene ridotta in modo comparativo. Frequent Pattern Tree è una struttura ad albero creata con i set di elementi iniziali del database. Lo scopo dell'albero FP è di estrarre il pattern più frequente, ogni nodo dell'albero FP rappresenta un elemento dell'insieme di elementi.

Il nodo radice rappresenta il “null” mentre i nodi inferiori rappresentano i set di elementi. L'associazione dei nodi con i nodi inferiori, ovvero gli insiemi di elementi con gli altri insiemi di elementi, viene mantenuta durante la formazione dell'albero.

Il metodo di crescita frequente del modello ci consente di trovare il modello frequente senza generazione di candidati.

I passaggi seguiti per minare il pattern frequente utilizzando l'algoritmo di crescita del pattern frequente sono i seguenti:

- 1) Il primo passo è eseguire la scansione del database per trovare le occorrenze dei set di elementi nel database. Questo passaggio è lo stesso del primo passaggio di Apriori. Il conteggio di 1 set di elementi nel database è chiamato conteggio supporto o frequenza di 1 set di elementi.
- 2) Il secondo passo è costruire l'albero FP. Per questo, crea la radice dell'albero. La radice è rappresentata da null.
- 3) Il passaggio successivo è eseguire nuovamente la scansione del database ed esaminare le transazioni. Esamina la prima transazione e scopri il set di elementi in essa contenuto. Il set di elementi con il conteggio massimo viene preso in alto, il set di elementi successivo con il conteggio inferiore e così via. Significa che il ramo dell'albero è costruito con insiemi di elementi di transazione in ordine decrescente di conteggio.
- 4) Viene esaminata la transazione successiva nel database. I set di elementi sono ordinati in ordine decrescente di conteggio. Se un set di elementi di questa transazione è già presente in un altro ramo (ad esempio nella prima transazione), questo ramo della transazione condividerà un prefisso comune alla radice. Ciò significa che l'insieme di elementi comune è collegato al nuovo nodo di un altro insieme di elementi in questa transazione.
- 5) Il conteggio dell'insieme di elementi viene incrementato man mano che si verifica nelle transazioni. Sia il nodo comune che il conteggio del nuovo nodo vengono aumentati di 1 man mano che vengono creati e collegati in base alle transazioni.
- 6) Il passaggio successivo consiste nel minare l'albero FP creato. Per questo, il nodo più basso viene esaminato per primo insieme ai collegamenti dei nodi più bassi. Il nodo più basso rappresenta la lunghezza del pattern di frequenza 1. Da questo, attraversare il percorso nell'albero FP. Questo percorso o percorsi sono chiamati base del modello condizionale.

La base del modello condizionale è un database secondario costituito da percorsi di prefisso nell'albero FP che si verificano con il nodo (suffisso) più basso.

7) Costruisci un albero FP condizionale, formato da un conteggio di set di elementi nel percorso. I set di elementi che soddisfano il supporto della soglia sono considerati nella struttura FP condizionale.

8) I pattern frequenti vengono generati dall'albero FP condizionale.

Algoritmo ECLAT

L'algoritmo ECLAT sta per Equivalence Class Clustering and bottom-up Lattice Traversal. È uno dei metodi più diffusi di estrazione delle regole di associazione. È una versione più efficiente e scalabile dell'algoritmo Apriori. Mentre l'algoritmo Apriori lavora in senso orizzontale, imitando la Breadth-First Search di un grafo, l'algoritmo ECLAT lavora in modo verticale, proprio come la Depth-First Search di un grafo. Questo approccio verticale dell'algoritmo ECLAT lo rende più veloce dell'algoritmo Apriori.

Come funziona l'algoritmo?

L'idea di base è quella di utilizzare le intersezioni dei Transaction Id Sets (tidsets) per calcolare il valore di supporto di un candidato, evitando di generare sottoinsiemi che non esistono nell'albero dei prefissi. Nella prima chiamata della funzione, vengono utilizzati tutti i singoli elementi insieme ai loro tidset. Poi la funzione viene richiamata in modo ricorsivo e in ogni chiamata ricorsiva, ogni coppia elemento-tidet viene verificata e combinata con altre coppie elemento-tidet. Questo processo continua fino a quando non è possibile combinare nessuna coppia item-tidset candidata.

Differenze tra Apriori e FP

Una delle caratteristiche più importanti di qualsiasi algoritmo di estrazione di insiemi di oggetti frequenti è che deve richiedere tempi e memoria ridotti. Tenendo conto di ciò, esistono molti algoritmi legati agli algoritmi FIM (Frequent Itemset Mining). I due algoritmi Apriori e FP-Growth sono gli algoritmi FIM di base. Altri algoritmi in questo campo sono miglioramenti di questi algoritmi. Esistono alcune differenze di base tra questi algoritmi, che vediamo qui di seguito. Per quanto riguarda la generazione di pattern l'algoritmo Apriori genera i modelli frequenti creando gli insiemi di elementi utilizzando accoppiamenti come single item set, doppio itemset, triplo itemset, mentre FP Growth genera un FP-Tree per creare modelli frequenti.

Apriori utilizza la generazione di candidati in cui i sottoinsiemi frequenti vengono estesi un elemento alla volta mentre FP-growth non utilizza la generazione di candidati. Un'altra differenza sostanziale la troviamo nella velocità con cui i due algoritmi scansionano i dataset, poiché apriori esegue la scansione del database in ciascuno dei suoi passaggi, diventa dispendioso in termini di tempo per i dati in cui il numero di elementi è maggiore, invece FP-tree richiede solo una scansione del database nelle sue fasi iniziali e quindi consuma meno tempo. Abbiamo delle differenze anche per quanto riguarda le modalità di salvataggio, per l'algoritmo apriori una versione convertita del database viene salvata nella memoria, mentre quando usiamo l'algoritmo FP viene salvata una versione compatta del database.

In conclusione, possiamo dire che L'algoritmo Apriori viene utilizzato per le regole di associazione mineraria. Funziona in base al principio, 'anche i sottoinsiemi non vuoti di insiemi di elementi frequenti devono essere frequenti'. Forma candidati k-itemset da (k-1) set di elementi ed esegue la scansione del database per trovare i set di elementi frequenti.

Frequent Pattern Growth Algorithm è il metodo per trovare pattern frequenti senza generazione di candidati. Costruisce un albero FP invece di utilizzare la strategia di

generazione e test di Apriori. L'obiettivo dell'algoritmo di crescita FP è sulla frammentazione dei percorsi degli elementi e sull'estrazione di modelli frequenti.

Algoritmi nella letteratura

Le numerose informazioni a disposizione delle aziende hanno bisogno di essere investigate dal punto di vista quantitativo attraverso una serie di algoritmi e metodi, che permettano di estrarre una conoscenza. L'articolo "Mapping of regulatory actors and processes related to cloud-based energy management environments using the Apriori algorithm" si pone l'obiettivo di identificare i processi e gli attori che compongono il livello di regolamentazione dei sistemi energetici, stabilire le relazioni di base tra questi attori e processi, delineando le linee guida per la creazione e/o la modifica di politiche, leggi e regolamenti relativi alla transizione dei sistemi di gestione dell'energia utilizzando una revisione sistematica della letteratura (SLR) e l'algoritmo Apriori. La SLR ha identificato 7 processi principali e 21 processi secondari, per un totale di 28 processi normativi (delineati e presentati attraverso una mappa mentale), stabilendo attraverso Apriori una rete di dipendenze tra questi processi con 37 collegamenti diretti. Sono stati identificati 23 attori che sono strutturati in una rete con 28 collegamenti diretti e dipendenti. Le connessioni tra i processi e gli attori, secondo l'articolo, serviranno come punto di partenza per la creazione di una roadmap per lo sviluppo di nuove normative che tengano conto dell'implementazione della CE.

In un altro scritto "Inferring implications in semantic maps via the Apriori algorithm" l'algoritmo apriori è stato utilizzato perché le mappe semantiche esistenti non sono in grado di illustrare lo squilibrio delle implicazioni tra le funzioni, il problema dell'inferenza delle implicazioni e dello sbilanciamento delle implicazioni tra le funzioni è simile al ben noto problema di generare tutte le regole di associazione significative tra gli articoli acquistati dai clienti, quindi l'algoritmo Apriori offre una soluzione efficace per risolvere questo problema.

L'algoritmo Apriori è stato molto utile per analizzare gli incidenti stradali, troviamo in "Apriori-Based Algorithm for Dubai Road Accident Analysis" uno studio che utilizza tecniche di data mining per analizzare i dati sugli incidenti stradali relativi a Dubai per l'anno 2017 e l'algoritmo Apriori è stato utilizzato per estrarre gli insiemi frequenti. È stato osservato che la maggior parte degli incidenti coinvolge un veicolo che tampona un altro veicolo a causa dello spazio inadeguato tra i veicoli. Un altro dato, da tener d'occhio, emerso è che i giovani sono coinvolti nella maggior parte degli incidenti. I risultati mostrano che gli incidenti sono più frequenti nella tarda notte e che la maggior parte dei conducenti è stata trovata in stato di ebbrezza. Gli studi condotti dimostrano che durante l'ora di punta degli incidenti nel fine settimana la maggior parte dei conducenti era in stato di ebbrezza, mentre durante l'ora di punta degli incidenti nei giorni feriali il numero massimo di incidenti si è verificato a causa della mancanza di spazio sufficiente tra i veicoli. Sulla base dei risultati ottenuti, sono state formulate alcune raccomandazioni che aiutano a ridurre il tasso di incidenti.

La previsione del rischio di disastri è uno dei modi più efficaci per prevenire gli incidenti, nell'articolo "Risk prediction and factors risk analysis based on IFOA-GRNN and apriori algorithms: Application of artificial intelligence in accident prevention" si propone un nuovo metodo per la previsione del rischio e l'analisi del rischio. Come oggetto di ricerca sono stati scelti gli incidenti dovuti all'esplosione di carbone e gas, dopo aver utilizzato algoritmo di ottimizzazione della mosca della frutta (IFOA) e la rete neurale di regressione generale (GRNN), questi sono stati combinati per stabilire il modello di previsione IFOA-GRNN e l'algoritmo Apriori è stato utilizzato per estrarre le informazioni sui disastri. In questo caso Apriori ha eseguito un buon data mining sulla combinazione dei fattori ad alta frequenza che portano agli incidenti e sulle relazioni tra i livelli di esplosione di carbone e gas e i fattori. I risultati del data mining si sono rivelati molto utili per la prevenzione e la gestione delle esplosioni di carbone e gas.

La tecnica di data mining Apriori è stata utilizzata anche per identificare, inerente alla situazione pandemica causata dal COVID-19, le combinazioni di variabili del paziente associate alla mortalità o alla sopravvivenza. Questo studio lo troviamo in “PIN108 Using the Apriori Algorithm to Identify Risk Factors Associated with Survival and Mortality Among COVID-19 Patients”. Per portare avanti questo studio è stata utilizzata l’analisi dei dati utilizzando le cartelle cliniche dei pazienti con diagnosi COVID-19 presenti nel Premier Healthcare Database, da settembre 2019 a maggio 2020. Le regole di associazione sono state generate utilizzando l’algoritmo Apriori, mantenendo il supporto minimo e la confidenza minima a 0,004 e 0,65 per l’esito della mortalità, e 0,10 e 0,95 per l’esito della sopravvivenza.

Leggendo l’articolo “Apriori and K-Means algorithms of machine learning for spatio-temporal solar generation balancing” veniamo a conoscenza che il numero di impianti solari fotovoltaici (FV) di grandi dimensioni collegati alla rete è aumentato in modo significativo negli ultimi 10 anni, con una conseguente elevata penetrazione dell’energia FV nella rete, soprattutto per i paesi a distribuzione spaziale ad ampio raggio, la rampa di potenza di un impianto fotovoltaico può essere bilanciata con la generazione di un altro impianto fotovoltaico. In questo studio le regole di associazione generate con l’algoritmo Apriori forniscono mappe di direzione della rampa di alimentazione per l’analisi spazio-temporale. Il clustering K-means (basato sull’algoritmo di Hartigan-Wong) è utilizzato per l’apprendimento non supervisionato delle relazioni spazio-temporali per le zone di rampa del solare fotovoltaico. Viene proposto un modello che può essere utilizzato come strumento decisionale rapido ed efficace con risultati qualitativi per l’operatore di sistema con conoscenze specialistiche minime, mentre può essere integrato nell’analisi dei vincoli del flusso di potenza ottimale. Questo studio introduce la prima applicazione di successo delle regole di associazione integrate con la tecnica K-means per il bilanciamento spazio-temporale del solare fotovoltaico, per quanto ne sappiamo.

In letteratura non troviamo soltanto scritti dove vengono proposti scenari e studi grazie all'aiuto dell'algoritmo Apriori, ma ci sono anche molteplici articoli dove vengono proposti casi di studio utilizzando l'algoritmo FP-growth. In "A heuristic approach for load balancing the FP-growth algorithm on MapReduce" si propone una strategia di bilanciamento del carico per l'algoritmo PFP, in quanto l'algoritmo di crescita dei pattern frequenti (FP), spesso utilizzato per la scoperta di insiemi frequenti, non è in grado di adattarsi direttamente ai Big Data di oggi, soprattutto per i grandi insiemi di dati sparsi. Gli autori hanno trovato necessario distribuire e parallelizzare l'algoritmo FP-growth. Parallel FP-growth (PFP) è un'implementazione parallela dell'algoritmo FP-growth sul framework di esecuzione MapReduce di Hadoop, sebbene PFP sia scalabile su grandi insiemi di dati, soffre di uno squilibrio di carico tra le unità di elaborazione, la strategia proposta per il bilanciamento del carico è basata su un'euristica e con un ordine di complessità inferiore, chiamata Heuristic Based PFP (HBFP). Alla fine i risultati hanno dimostrato che HBFP distribuisce il carico in modo più uniforme tra i nodi del cluster Hadoop, funziona più velocemente dell'algoritmo PFP e utilizza le risorse del cluster in modo più efficiente, soprattutto per i grandi insiemi di dati sparsi.

Possiamo trovare spesso anche proposte di nuovi algoritmi per l'estrazione mirata di più insiemi, e di questo viene discusso in "A guided FP-Growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data". Gli autori hanno proposto un nuovo algoritmo per l'estrazione mirata di più insiemi, denominato Guided Frequent Pattern-Growth (GFP-Growth). L'algoritmo GFP-Growth è stato progettato per estrarre rapidamente un dato insieme di item utilizzando una piccola quantità di memoria. Questo algoritmo è stato proposto in quanto trovare tutti gli insiemi di elementi frequenti in insiemi di dati grandi o densi può richiedere molto tempo e un utente può essere interessato solo ad alcuni insiemi di elementi specifici piuttosto che a tutti. Nel manoscritto è stato dimostrato che GFP-Growth produce i conteggi di frequenza esatti per ogni insieme di elementi di interesse, si dimostra inoltre che GFP-Growth può aumentare le prestazioni di diversi

problemi che richiedono l'estrazione di insiemi di elementi. In particolare, hanno studiato il problema della generazione di regole di classe minoritaria da dati sbilanciati e sviluppiamo l'algoritmo Minority-Report (MRA) che utilizza GFP-Growth per risolvere questo problema in modo efficiente.

E' molto interessante ciò che viene affrontato in "Map-optimize-reduce: CAN tree assisted FP-growth algorithm for clusters based FP mining on Hadoop". Qui troviamo una soluzione diversa per ovviare il problema degli ultimi anni, cioè, è vero che l'estrazione di pattern frequenti (FPM) sta emergendo come un approccio significativo per scoprire l'affascinante conoscenza nascosta nei dati, tuttavia, i lavori precedenti non hanno affrontato la validazione di FPM con le interrogazioni degli utenti e anche il raggiungimento di una migliore scalabilità e il tempo di esecuzione è ancora un collo di bottiglia a causa delle difficoltà nel gestire grandi set di dati. La proposta di lavoro di questo articolo stabilisce FPM utilizzando una versione estesa del framework MapReduce in ambiente Hadoop. Il lavoro che è stato proposto esegue una pre elaborazione per rimuovere la ridondanza dei dati. Per accelerare il framework MapReduce, hanno usato il clustering AP che genera cluster efficaci dal set di dati dato. Il bilanciamento del carico viene eseguito per bilanciare il carico tra i diversi blocchi in cui viene calcolata la reputazione e per evitare sviste nella scansione e minimizzare lo spazio di ricerca in MapReduce, tra Mapper e Reducer hanno inserito un ottimizzatore che utilizza l'ottimizzazione Emperor Penguin Colony (EPC). L'utente fornisce Mining_Request ad Hadoop e i modelli frequenti vengono estratti per la richiesta data che viene rispedita all'utente. Se la richiesta data dall'utente non è presente nell'albero CAN, il sistema invia un Relevance Feedback come raccomandazione all'utente.

L'algoritmo FP-growth è stato utilizzato, come leggiamo in "FP-Growth based Regular Behaviors Auditing in Electric Management Information System", per evitare di commettere errori che possono intaccare la sicurezza di un sistema. In questo caso si parla di sistemi informativi gestionali (MIS) elettrici, qui possiamo

trovare alcuni utenti che non rispettano tutte le regole di funzionamento o comportamento e commettono continuamente errori simili, anche se non di proposito, questi comportamenti rappresentano un'enorme minaccia per la sicurezza del sistema. Un metodo per rilevare questi comportamenti regolari può essere quello di usare l'algoritmo di estrazione di regole di associazione FP-Growth. Per avanzare questo studio, in primo luogo il registro degli utenti viene separato in insiemi di operazioni, ciascuno dei quali contiene operazioni dell'utente in un periodo continuo. Poi sono stati divisi gli insiemi di operazioni di tutti gli utenti in due cataloghi: normale e anormale, in base al fatto che si sia verificato un problema di sicurezza nel periodo corrispondente dell'insieme di operazioni. Successivamente, è stato applicato l'algoritmo FP-Growth a entrambi gli insiemi di operazioni normali e anormali per generare i pattern frequenti. Infine, i pattern anomali vengono confrontati con quelli normali per determinare i comportamenti regolari che possono essere pericolosi per il sistema. Alla fine i risultati dell'esperimento indicano che il metodo proposto è in grado di rilevare efficacemente i comportamenti regolari degli utenti che potrebbero causare problemi di sicurezza al sistema.

Non solo l'algoritmo Apriori, ma anche quello FP-growth è stato usato in ambito medico per la previsione di malattie potenzialmente letali come il cancro, la cirrosi epatica e l'epatite, come viene discusso in "Identification of adverse disease agents and risk analysis using frequent pattern mining". Nel testo si parla di come il rischio di mortalità può essere ridotto grazie alla diagnosi precoce dei sintomi e alla fornitura della migliore diagnosi possibile. Questo ruolo critico di rilevamento e/o diagnosi può essere potenziato utilizzando una delle tecniche di data mining, come il pattern mining periodico, l'association rule mining e la classificazione. Verso l'adozione dell'estrazione di regole di associazione, l'ideazione di un algoritmo economico e che consenta di risparmiare tempo per l'estrazione di pattern frequenti riveste un ruolo importante. In questo lavoro è stato proposto un approccio all'estrazione di pattern chiamato Improved Frequent Pattern Growth (Improved FP-Growth). In primo luogo, è stata costruita una struttura dati ad albero di pattern frequenti improvvisata,

chiamata Improved FP-tree. Inoltre, Improved FP-Growth introduce una struttura dati condizionale FP-tree chiamata Improved Conditional Frequent Pattern Tree (Improved Conditional FP-Tree). A differenza del metodo FP-Growth tradizionale, utilizza approcci sia top-down che bottom-up per generare in modo efficiente modelli frequenti senza costruire ricorsivamente l'albero FP condizionale migliorato. I risultati sperimentali hanno sottolineato l'importanza dell'algoritmo Improved FP-Growth proposto rispetto ad alcuni algoritmi tradizionali di frequent itemset mining che adottano l'approccio della costruzione ricorsiva dell'FP-tree condizionale. I risultati sperimentali hanno mostrato che l'algoritmo di costruzione di FP-tree migliorato supera l'algoritmo di costruzione di FP-tree convenzionale in termini di tempo di esecuzione in tutti i casi, sia per i database radi che per quelli densi. D'altra parte, dai risultati sperimentali basati sul tempo di estrazione, si può osservare che per i database sparsi l'algoritmo Improved FP-Growth ha superato gli algoritmi esistenti. Pertanto, l'algoritmo di estrazione di itemset frequenti proposto ha raggiunto un buon compromesso tra tempo di estrazione e requisiti di memoria per la generazione di modelli frequenti.

Capitolo 4

Introduzione e obiettivo

Il seguente lavoro di tesi si concentra sullo sviluppo di un tool di intelligenza artificiale che permetta di assegnare regole di associazione in nell'azienda di ceramiche MATRIX Srl, per la programmazione della produzione. Le regole di associazione, come già detto nei capitoli precedenti, sono una tecnica di analisi dei dati che utilizza l'algoritmo di associazione per identificare modelli nascosti nei dati. Questi modelli sono utili per prevedere il comportamento delle vendite e delle preferenze dei clienti, ma in un'azienda di ceramiche, la conoscenza dei modelli nascosti nei dati può aiutare a prendere decisioni sulla produzione e la distribuzione di prodotti. Tuttavia, l'analisi manuale dei dati può essere complessa e richiedere molto tempo per questo è stato necessario sviluppare un codice che utilizza l'algoritmo di associazione per automatizzare l'analisi dei dati e assegnare regole di associazione. Il codice è stato sviluppato utilizzando Python e la libreria mlxtend. La tesi descrive in dettaglio il processo di sviluppo del codice, le sfide incontrate durante il processo e i risultati ottenuti, con l'obiettivo di fornire una soluzione pratica e automatizzata per l'analisi dei dati e l'assegnazione di regole di associazione nell'azienda presa in esame.

Lo sviluppo di questo codice è solamente una piccola parte dell'intero sistema ideato per l'azienda MATRIX, infatti il tool di intelligenza artificiale prende i dati da archivi storici, ottiene le regole e successivamente queste vengono passate ad uno schedulatore che programmerà l'intera produzione. Quanto appena descritto è mostrato schematicamente nell'immagine 1.

AIDEAS
Fabrication
Optimiser
FRAMEWORK

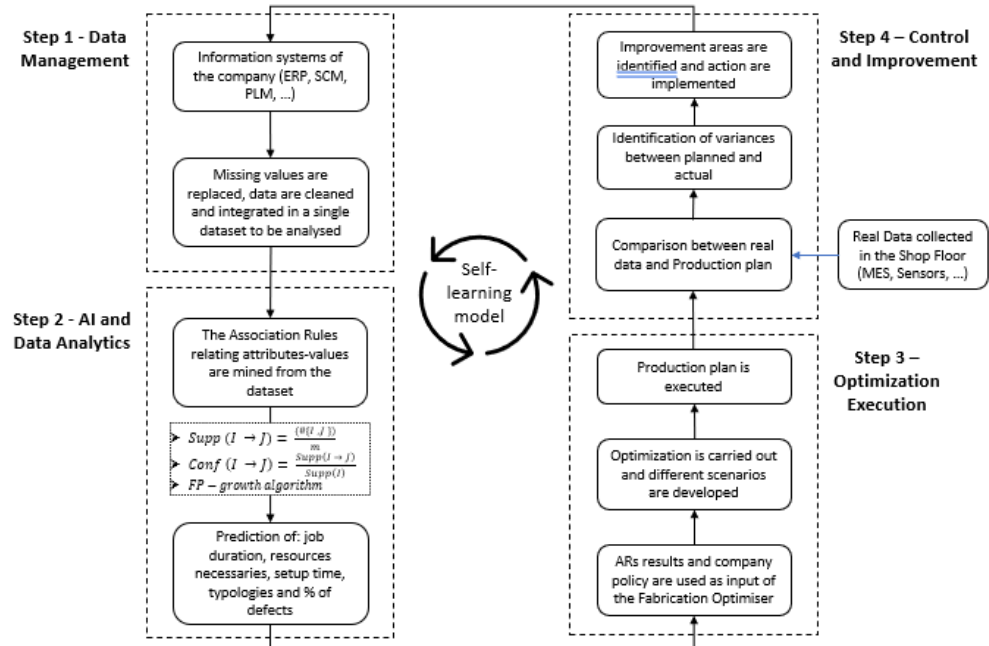


Immagine 1

Librerie che implementato algoritmi di associazione

Il primo passo è stato quello di trovare una libreria Python per il machine learning che fornisce una serie di estensioni e funzionalità utili per la data science e l'analisi dei dati, nel caso specifico una libreria capace di fornire regole di associazione. Come già accennato in precedenza la libreria usata è stata Mlxtend, ma prima di arrivare a questa sono state testate le librerie Apriori e Pyfpgrowth. Per scegliere la libreria appropriata al caso di studio, tutte e tre sono state testate con un insieme di dati trovato su internet. L'insieme rappresentava vari oggetti acquistati da diversi clienti, come si può vedere nell'immagine 2.

	A	B	C	D	E	F	G	H	I
1	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink
2	burgers	meatballs	eggs						
3	chutney								
4	turkey	avocado							
5	mineral water	milk	energy bar	whole wheat rice	green tea				
6	low fat yogurt								
7	whole wheat pasta	french fries							
8	soup	light cream	shallot						
9	frozen vegetables	spaghetti	green tea						
10	french fries								
11	eggs	pet food							
12	cookies								
13	turkey	burgers	mineral water	eggs	cooking oil				

Immagine 2

Gli oggetti sono organizzati come elementi di un elenco, dove ogni riga rappresenta un singolo acquisto. Gli elementi vuoti indicano che un certo prodotto non è stato acquistato durante quel particolare acquisto, il dataset è stato scelto in quanto ritenuto adatto per essere utilizzato per analizzare le relazioni tra gli elementi acquistati e quindi utile per identificare tendenze o associazioni tra i prodotti. L'immagine riporta solo una parte del dataset che è composto da 7501 righe (transazioni) e un massimo di 20 oggetti per riga. La numerosità degli elementi è stato un fattore determinante per la scelta del dataset, in quanto garantiva la funzionalità, o meno, delle tre librerie.

La prima libreria testata è stata Apriori e il codice, sviluppato per una semplice ricerca delle regole di associazione, è riportato qui sotto (immagine 3).

```

1 import pandas as pd
2 from apyori import apriori
3 store_data = pd.read_csv(r"C:\Users\anton\Downloads\store_data.csv", header=None)
4 store_data.head()
5 records = []
6 for i in range(0, 7501):
7     records.append([str(store_data.values[i,j]) for j in range(0, 20)])
8 association_rules = list(apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=2))
9 print(association_rules[0])
10 for item in association_rules:
11     pair = item[0]
12     items = [x for x in pair]
13     print("Rule: " + items[0] + " -> " + items[1])
14     print("Support: " + str(item[1]))
15     print("Confidence: " + str(item[2][0][2]))
16     print("Lift: " + str(item[2][0][3]))
17     print("=====")

```

Immagine 3

Il codice di per sé è molto semplice, il file "store_data.csv", contenente le transazioni viste nell'immagine 2, viene caricato come dataframe con pandas (libreria di Python

per l'elaborazione e l'analisi dei dati) e ogni riga viene convertita in una lista di stringhe attraverso la funzione presente nella riga 7. La riga 8 permette di trovare le regole di associazione, tramite l'algoritmo apriori, andando ad impostare manualmente i parametri di supporto e confidenza. I comandi presenti dalla riga 10 alla riga 17 permettono di stampare i risultati delle regole di associazione generate dalla libreria apriori con i relativi parametri. Il codice si è rivelato molto affidabile in termini di tempo e veridicità dei risultati, ma il problema evidente è la marcata personalizzazione dello stesso, infatti la parte finale del codice, che stampa i risultati finali, deve essere cambiato per ogni dataset inserito e richiederebbe un lavoro molto più dispendioso in termini di tempo all'interno dell'azienda presa in esame. Un altro problema riscontrato, che ha portato a scartare questa libreria, è il fatto che riusciva a fornire solamente antecedente e conseguente singoli. I problemi appena descritti hanno portato a cercare e provare un'altra libreria, in questo caso la seconda libreria provata è stata la libreria Pyfpgrowth.

Quest'ultima implementa l'algoritmo FP-growth per la scoperta di frequent itemset, questo algoritmo è efficiente per la scoperta di frequent itemset che utilizza una struttura di dati chiamata albero FP per comprimere i dati di input e ridurre la quantità di calcoli necessari per scoprire gli itemset frequenti. Il vantaggio della libreria pyfpgrowth è l'utilizzo di un'interfaccia semplice per utilizzare l'algoritmo FP-growth e utilizzare i risultati per scoprire regole di associazione. Il codice sviluppato è riportato nell'immagine 4.

```
1 import pandas as pd
2 import numpy as np
3 import pyfpgrowth
4 store_data = pd.read_csv(r"C:\Users\anton\Downloads\store_data.csv", header=None)
5 records = []
6 for i in range(0, 10):
7     records.append([str(store_data.values[i,j]) for j in range(0, 20)])
8
9
10 patterns = pyfpgrowth.find_frequent_patterns(records, 2)
11 for item in patterns:
12     rules = pyfpgrowth.generate_association_rules(item, .7)
```

Immagine 4

Questo script, attraverso la libreria appena descritta, analizza un insieme di dati memorizzato in un file CSV (transazioni citate in precedenza). Viene utilizzato il metodo `find_frequent_patterns` per identificare i pattern frequenti all'interno dei dati. I pattern frequenti vengono quindi utilizzati per generare regole di associazione utilizzando il metodo `generate_association_rules`. Nonostante a prima vista il codice sia molto semplice, le problematiche sono state riscontrate nella determinazione dei risultati. Per consentire al codice di fornire i risultati in un periodo di tempo ragionevole, bisognava creare un file con un massimo di 20 elementi, o transazioni, dunque un numero che non consentiva di riscontrare la veridicità dei risultati. Testandolo con l'intero file, 7501 transazioni, il codice forniva i risultati delle regole di associazione dopo almeno 20 minuti, un tempo troppo elevato per lo scopo del progetto. Le varie problematiche riscontrate con l'utilizzo delle librerie appena descritte, hanno portato alla scelta della libreria `Mlxtend` che verrà descritta nei capitoli successivi dove si parlerà dell'analisi effettuata per l'azienda di ceramiche MATRIX.

Sviluppo del tool

In questo paragrafo verrà descritto tutto il procedimento che ha portato allo sviluppo del codice usando il linguaggio di programmazione python, utilizzando principalmente la libreria `Mlxtend`, per ottenere delle regole di associazione da sfruttare successivamente per la programmazione della produzione nell'azienda Matrix. Prima di descrivere nel dettaglio il codice ed il suo funzionamento, verrà mostrato e brevemente descritto il dataset fornito dall'azienda per il progetto di questa tesi.

L'immagine 5 e l'immagine 6 mostrano una breve porzione dell'intero set fornito.

Tipo	data-5	data-4	Data	Articolo
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Azi Lavabo
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Azi Wc
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Biis
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Blat
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Chim
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Clo 70
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Dudu 650
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Juno bide
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Juno Wc
Prime cotture	02/05/2020	03/05/2020	07/05/2020	UB 850 Wc
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Ulm 60
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Apron 30"
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Apron 33"
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Apron 36"
Prime cotture	02/05/2020	03/05/2020	07/05/2020	Orchard 36"

Immagine 5

Buoni	%Buoni	Ripasso	%Ripasso	Scarto	%Scarto	2a	%2a	NOTE	mettere il tipo di impasto_t	mettere il tipo di impasto_t
1	100%		0%		0%		0%		0	0
1	33%	1	33%	1	33%		0%		0	0
3	75%	1	25%		0%		0%		0	0
	0%	1	100%		0%		0%		0	0
2	100%		0%		0%		0%		0	0
1	100%		0%		0%		0%		0	0
1	100%		0%		0%		0%		0	0
	0%	1	50%	1	50%		0%		0	0
1	50%	1	50%		0%		0%		0	0
1	100%		0%		0%		0%		0	0
3	100%		0%		0%		0%		0	0
4	80%	1	20%		0%		0%		0	0
2	67%	1	33%		0%		0%		0	0
	0%	1	100%		0%		0%		0	0
3	43%	1	14%	3	43%		0%		0	0
4	80%		0%	1	20%		0%	FFC Imerys	0	0

Immagine 6

Queste immagini rappresentano una tabella excel che presenta i risultati di un controllo di qualità su vari articoli di ceramica. La tabella ha una struttura riga per riga, dove ogni riga rappresenta un articolo, tutti diversi tra loro, identificati dalla colonna "Articolo". Per ogni articolo troviamo, come mostrato dall'immagine 7, diverse tipologie di difettosità insieme alla loro numerosità, ovviamente la cella vuota indica che non è stato riscontrato quel tipo di difetto.

Articolo	1. Spillatura	2. Punti color	3. Grana/ fibi	4. Crepa interna	5. Cre	6. Crep	7. Difetti smalto	8. Macchie verdi	9. Defo	10. Log
Azi Lavabo										
Azi Wc							1			
Biis								1		
Blat		1								
Chim										
Clo 70	1									
Dudu 650										
Juno bidet						N		X		
Juno Wc	1									
UB 850 WH										
Ulm 60										
Apron 30"							1			
Apron 33"							1			
Apron 36" SB							1			
Orchard 36 "				XXX						1
Biis				X						
Blat				X		X1				

Immagine 7

Inoltre, sono presenti colonne che mostrano la quantità di articoli che sono stati considerati buoni, da ripassare o da scartare e questi sono identificati dalle colonne "Buoni", "Ripasso", "Scarto", con la rispettiva percentuale presente nelle colonne "%Buoni", "%Ripasso", "%Scarto" (immagine6). Le colonne successive danno informazioni sulla tipologia di impasto usata per ogni articolo, nell'immagine mostrata si nota solamente una tipologia "FFC Imerys", ma nell'intero dataset sono presenti anche le tipologie "VC Euroarce" e "VC Imerys"; tutte e tre possono essere utilizzate sia singolarmente che insieme, non abbiamo informazione sulla tipologia di impasto se troviamo lo spazio vuoto o lo 0 al di sotto di queste colonne.

Partendo da questo dataset l'obiettivo prefissato è stato quello di trovare regole di associazione in presenza di informazioni sulla tipologia di articolo, sul tipo di cottura, sul periodo dell'anno in cui veniva fabbricato l'articolo, in quanto la temperatura nei diversi periodi poteva influire sul numero di pezzi privi di difetti, sulla tipologia di ceramica, sul numero di pezzi infornati e queste informazioni sono state utilizzate come input, invece come output sono state usate le informazioni relative al numero di pezzi aventi difettosità nulla. È stato necessario, partendo da questo file, creare un file csv che contenesse queste informazioni, così da poterlo inserire nel codice e estrarre le regole di associazione di nostro interesse.

Trovando molto difficile l'aver lo stesso numero di informazioni per ogni articolo, ma solamente quelle relative alla tipologia di cottura e al periodo in cui è stato

informato l'articolo, è stato creato un file CSV a 3 livelli. Nel primo livello troviamo tutte le tipologie di informazioni, utilizzate come input, relative agli articoli, nel secondo livello troviamo sempre il tipo di cottura e il periodo di infornata variando a due a due l'articolo, il numero di pezzi e la ceramica. Nel terzo e ultimo livello le informazioni fisse sono sempre presenti e sono collegate una volta con la tipologia di articolo, una volta con la tipologia di ceramica e una volta con il numero di pezzi informati. In ogni livello il dato utilizzato come output è stata la percentuale di pezzi privi di difettosità. È stata fatta questa divisione in livelli ragionando sul numero e la tipologia di dati che può avere l'azienda relative ad un'infornata in un periodo dell'anno. Per esempio se nel periodo di Maggio dell'anno precedente si hanno informazioni solamente sulla tipologia di ceramica utilizzata per ogni articolo, ma non si conosce l'articolo e nemmeno il numero di pezzi informati, per trovare le regole di associazione che mi permettono di programmare la produzione di Maggio, dovrò utilizzare i dati presenti nel livello 3 e inserirli nel codice scritto in python. Questa impostazione a livelli è resa più chiara dall'immagine 8.

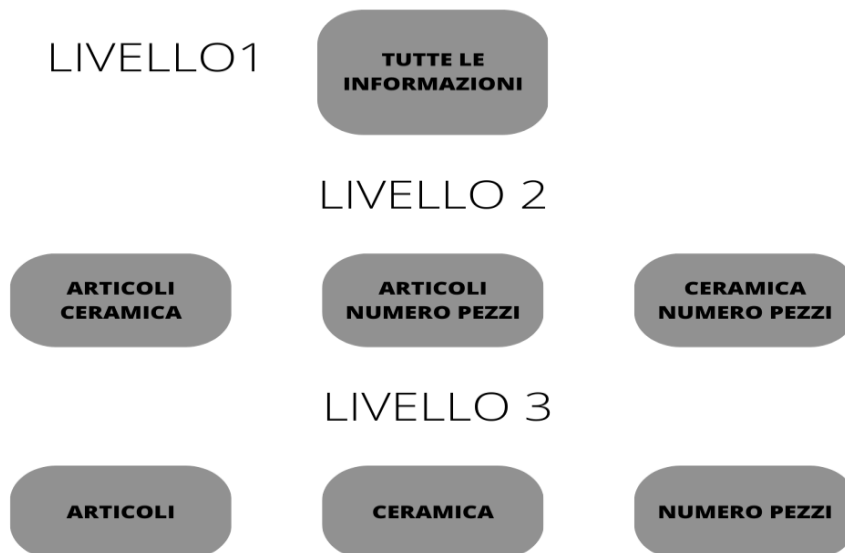


Immagine 8

Fatta questa breve descrizione del dataset utilizzato come input, andiamo a vedere il codice pensato per l'azienda Matrix. Le immagini successive mostrano lo script del codice, come editor di codice è stato usato, come per la prova delle librerie precedenti, Visual Studio Code.

```
1 import pandas as pd
2 import numpy as np
3 from mlxtend.preprocessing import TransactionEncoder
4 from mlxtend.frequent_patterns import apriori
5 from mlxtend.frequent_patterns import association_rules
6 import openpyxl as op
7 from openpyxl import load_workbook
8 from openpyxl import Workbook
9 from funzione import lett_input, algoritmo
10
11
12 nome_file= load_workbook(filename='Input_codice.xlsx', read_only=True)
13 nome_foglio1=nome_file['LVL1']
14 casoLVL1= lett_input(nome_file, nome_foglio1)
15 nome_foglio2=nome_file['LVL2a+c']
16 casoLVL2a_c = lett_input(nome_file,nome_foglio2)
17 nome_foglio3= nome_file['LVL2a+np']
18 casoLVL2a_np = lett_input(nome_file, nome_foglio3)
19 nome_foglio4 = nome_file['LVL2np+c']
20 casoLVL2c_np = lett_input(nome_file, nome_foglio4)
21 nome_foglio5 = nome_file['LVL3a']
22 casoLVL3a = lett_input(nome_file, nome_foglio5)
23 nome_foglio6 = nome_file['LVL3c']
24 casoLVL3c = lett_input(nome_file, nome_foglio6)
25 nome_foglio7 = nome_file['LVL3np']
```

Immagine 9

```
27 file_caso1 = algoritmo(casoLVL1)
28 file_caso2 = algoritmo(casoLVL2a_c)
29 file_caso3 = algoritmo(casoLVL2a_np)
30 file_caso4 = algoritmo(casoLVL2c_np)
31 file_caso5 = algoritmo(casoLVL3a)
32 file_caso6 = algoritmo(casoLVL3c)
33 file_caso7 = algoritmo(casoLVL3np)
34 with pd.ExcelWriter('analisi.xlsx') as writer:
35     file_caso1.to_excel(writer, sheet_name="LVL1")
36     file_caso2.to_excel(writer, sheet_name="LVL2a_c")
37     file_caso3.to_excel(writer, sheet_name="LVL2a_np")
38     file_caso4.to_excel(writer, sheet_name="LVL2c_np")
39     file_caso5.to_excel(writer, sheet_name="LVL3a")
40     file_caso6.to_excel(writer, sheet_name="LVL3c")
41     file_caso7.to_excel(writer, sheet_name="LVL3np")
42
43 nome_file = load_workbook(filename='analisi.xlsx', read_only=True)
44
```

Immagine 10

```

45 from Cod_antonio import filtro_analisi
46 nome_foglio1 = nome_file['LVL1']
47 nome_foglio2 = nome_file['LVL2a_c']
48 nome_foglio3 = nome_file['LVL2a_np']
49 nome_foglio4 = nome_file['LVL2c_np']
50 nome_foglio5 = nome_file['LVL3a']
51 nome_foglio6 = nome_file['LVL3c']
52 nome_foglio7 = nome_file['LVL3np']
53
54 file1 = filtro_analisi(nome_foglio1)
55 df1 = pd.DataFrame(file1)
56 df1 = df1.transpose()
57 file2 = filtro_analisi(nome_foglio2)
58 df2 = pd.DataFrame(file2)
59 df2 = df2.transpose()
60 file3 = filtro_analisi(nome_foglio3)
61 df3 = pd.DataFrame(file3)
62 df3 = df3.transpose()
63 file4 = filtro_analisi(nome_foglio4)
64 df4 = pd.DataFrame(file4)
65 df4 = df4.transpose()

```

Immagine 11

```

64 df4 = pd.DataFrame(file4)
65 df4 = df4.transpose()
66 file5 = filtro_analisi(nome_foglio5)
67 df5 = pd.DataFrame(file5)
68 df5 = df5.transpose()
69 file6 = filtro_analisi(nome_foglio6)
70 df6 = pd.DataFrame(file6)
71 df6 = df6.transpose()
72 file7 = filtro_analisi(nome_foglio7)
73 df7 = pd.DataFrame(file6)
74 df7 = df7.transpose()
75
76 with pd.ExcelWriter('analisifiltrata.xlsx') as writer:
77     df1.to_excel(writer, sheet_name="LVL1")
78     df2.to_excel(writer, sheet_name="LVL2a_c")
79     df3.to_excel(writer, sheet_name="LVL2a_np")
80     df4.to_excel(writer, sheet_name="LVL2c_np")
81     df5.to_excel(writer, sheet_name="LVL3a")
82     df6.to_excel(writer, sheet_name="LVL3c")
83     df7.to_excel(writer, sheet_name="LVL3np")
84

```

Immagine 12

Questo script analizza un insieme di dati forniti in un file excel "Input_codice.xlsx", che contiene 7 fogli corrispondenti ai 7 dataset precedentemente descritti, un primo dataset contenente tutte le informazioni e che quindi si trova al LIVELLO 1, tre dataset al LIVELLO 2 e tre al LIVELLO 3. I dataset sono composti da dati in input separati dalla “,” dall’unico dato in output, il numero dei pezzi non difettosi. . Il file excel, quindi, contiene sette fogli, ognuno dei quali viene elaborato con la funzione "lett_input" e successivamente con la funzione "algoritmo".

La funzione `lett_input` prende in input il nome del file e il nome del foglio e legge i dati riga per riga, suddividendo ogni riga in diverse colonne in base alla presenza del carattere ",". Questi dati vengono poi inseriti in una lista chiamata `store_data`.

La funzione `algoritmo` elabora i dati della lista `store_data` utilizzando l'algoritmo apriori per identificare gli insiemi frequenti degli item che si trovano nella stessa lavorazione. Questo algoritmo viene implementato dalla libreria `mlxtend`, così da calcolare successivamente l'insieme delle regole di associazione. Ottenute le regole di associazione, vengono selezionate solo quelle con una determinata soglia di confidenza, impostata a 0.0, così da averle tutte. Questi risultati vengono infine restituiti come un dataframe pandas (struttura di dati bidimensionale utilizzato per la manipolazione e l'analisi dei dati in Python). Lo script delle funzioni `lett_input` e `algoritmo` lo troviamo nelle immagini seguenti.

```
1 from openpyxl import load_workbook
2 import pandas as pd
3 import numpy as np
4 from mlxtend.preprocessing import TransactionEncoder
5 from mlxtend.frequent_patterns import apriori
6 from mlxtend.frequent_patterns import association_rules
7 def lett_input(nome_file, nome_foglio):
8     store_data=[]
9     line=[]
10    for row in nome_foglio.rows:
11        for cell in row:
12            line.append(cell.value)
13            store_data.append(list(line[0].split(",")))
14            line=[]
15    return (store_data)
```

Immagine 13

```

19 def algoritmo(store_data):
20
21     te = TransactionEncoder()
22     data_tr = te.fit(store_data).transform(store_data)
23     df = pd.DataFrame(data_tr, columns=te.columns_)
24     frequent_itemset = apriori(df, min_support= 0.005, use_colnames=True)
25     res = association_rules(frequent_itemset, metric="confidence", min_threshold=0.00)
26     res1 = res[["antecedents", "consequents", "support", "confidence", "lift"]]
27     res2 = res1[res1["confidence"] > 0.0]
28     return(res2)

```

Immagine 14

Qui sotto invece troviamo un'immagine relativa ad un foglio del file excel "Input_codice.xlsx". Vengono mostrate le informazioni che si possono avere in uno dei tre casi del LIVELLO 3, ossia le due fisse tipologia di cottura e periodo di informata e in questo caso la terza informazione è la tipologia di articolo. Possiamo vedere che i tre dati di input sono separati dal dato di output, numero pezzi non difettosi, dalla “,”.

	A	B	C	D
1	Prime cotture/Maggio/Bel,100%			
2	Prime cotture/Maggio/Biis,100%			
3	Prime cotture/Maggio/Blat,067%			
4	Prime cotture/Maggio/Bobo S/T,000%			
5	Prime cotture/Maggio/Chim,100%			
6	Prime cotture/Maggio/Clo 70,100%			
7	Prime cotture/Maggio/Dudu 650,050%			
8	Prime cotture/Maggio/Ulm 60,050%			
9	Prime cotture/Maggio/Apron 30",067%			
10	Prime cotture/Maggio/Apron 33",000%			
11	Prime cotture/Maggio/Apron 36" SB,000%			
12	Prime cotture/Maggio/Orchard 36 ",050%			
13	Prime cotture/Maggio/Azi Lavabo,100%			
14	Prime cotture/Maggio/Azi Wc,033%			
15	Prime cotture/Maggio/Biis,075%			
16	Prime cotture/Maggio/Blat,000%			
17	Prime cotture/Maggio/Chim,100%			
18	Prime cotture/Maggio/Clo 70,100%			
19	Prime cotture/Maggio/Dudu 650,100%			
20	Prime cotture/Maggio/Juno bidet,000%			
21	Prime cotture/Maggio/Juno Wc,050%			
22	Prime cotture/Maggio/UB 850 WH,100%			

Immagine 15

Continuando con la descrizione del codice, si può vedere dalle immagine 10, 11 e 12 come agisce lo stesso. In questi blocco viene utilizzato per analizzare e filtrare dati in formato Excel, utilizzando la libreria Pandas per manipolare i dati. È stata creata appositamente la funzione “filtro_analisi” che filtra i risultati di un’analisi effettuata tramite l'utilizzo di un algoritmo di associazione dei modelli (association rule mining). La funzione prende come input un foglio di lavoro, nel nostro caso i fogli dei vari LIVELLI, che rappresentano i risultati dell'analisi. La funzione legge i dati da questo foglio di lavoro, estraendo i valori delle celle in 5 liste: antecedente, conseguente, supporto, confidenza e lift. Dalla riga 34 il codice crea un nuovo file Excel chiamato "analisi.xlsx" e scrive sette fogli di dati (file_caso1, file_caso2, ..., file_caso7) in questo file con i nomi di foglio "LVL1", "LVL2a_c", ..., "LVL3np". Nella riga 43, viene caricato il file "analisi.xlsx" in modalità di sola lettura utilizzando la libreria openpyxl e crea sette oggetti (nome_foglio1, nome_foglio2, ..., nome_foglio7) che rappresentano i sette fogli di dati. Nelle riga 76 il codice utilizza la funzione "filtro_analisi" importata da "Cod_antonio" per filtrare i dati in ognuno dei sette oggetti e creare sette nuovi oggetti di DataFrame (df1, df2, ..., df7) che rappresentano i dati filtrati. L’ultimo blocco di codice crea un nuovo file Excel chiamato "analisifiltrata.xlsx" (immagine 16) e scrive i sette oggetti di DataFrame filtrati (df1, df2, ..., df7) in questo file come fogli di dati con i nomi di foglio "LVL1", "LVL2a_c", ..., "LVL3np". Questo ultimo file Excel rappresenta il risultato finale di questo progetto, un file contenente tutte le regole di associazione per i 7 file creati all’inizio dei 3 diversi LIVELLI.

	A	B	C	D	E	F
1		antecedente	conseguente	supporto	confidenza	lift
2	0	["Prime cotture", 'Maggio', 'Apron 33"', 'FFC Imerys', 'NP1']	0	0,006369	1	7,136364
3	1	["Prime cotture", 'Maggio', 'Apron 33"', 'NC', 'NP1']	0	0,006369	0,5	3,568182
4	2	["Prime cotture", 'Maggio', 'Apron 33"', 'NC', 'NP2']	0	0,006369	1	7,136364
5	3	["Prime cotture", 'Maggio', 'Apron 33"', 'VC Euroarce+FFC Imerys', 'NP1']	0	0,006369	0,5	3,568182
6	4	["Prime cotture", 'Maggio', 'Apron 36" SB', 'NC', 'NP1']	0	0,019108	1	7,136364
7	5	["Prime cotture", 'Maggio', 'Azi Bidet', 'VC Euroarce+FFC Imerys', 'NP0']	0	0,006369	1	7,136364
8	6	["Prime cotture", 'Maggio', 'Blat', 'NC', 'NP1']	0	0,012739	0,666666667	4,757576
9	7	["Prime cotture", 'Maggio', 'Bobo S', 'T', 'NC', 'NP1']	0	0,006369	1	7,136364
10	8	["Prime cotture", 'Maggio', 'Chim', 'NC', 'NP2']	0	0,006369	0,333333333	2,378788
11	9	["Prime cotture", 'Maggio', 'Clo 38', 'VC Euroarce+FFC Imerys', 'NP2']	0	0,006369	1	7,136364
12	10	["Prime cotture", 'Maggio', 'Clo 70', 'NC', 'NP3']	0	0,006369	1	7,136364
13	11	["Prime cotture", 'Maggio', 'Cono Liscio', 'FFC Imerys', 'NP1']	0	0,006369	1	7,136364
14	12	["Prime cotture", 'Maggio', 'Ger', 'VC Euroarce+FFC Imerys', 'NP1']	0	0,006369	1	7,136364
15	13	["Prime cotture", 'Maggio', 'Juno bidet', 'FFC Imerys+VC Imerys', 'NP2']	0	0,006369	1	7,136364
16	14	["Prime cotture", 'Maggio', 'Juno bidet', 'NC', 'NP2']	0	0,006369	1	7,136364
17	15	["Prime cotture", 'Maggio', 'Ode', 'FFC Imerys+VC Imerys', 'NP1']	0	0,006369	1	7,136364
18	16	["Prime cotture", 'Maggio', 'Orchard 36"', 'FFC Imerys', 'NP3']	0	0,006369	1	7,136364
19	17	["Prime cotture", 'Maggio', 'Zazu 600', 'FFC Imerys+VC Imerys', 'NP1']	0	0,006369	0,5	3,568182

Immagine 16

Conclusione

Una volta ottenute queste regole di associazione è necessario utilizzarle per fornire al programmatore della produzione, il numero di articoli da produrre nel momento in cui arriva un ordine da parte di un cliente. Il motivo per cui è stato utilizzato come dato di output la percentuale dei pezzi privi di difettosità, è perché conoscere quali tra questi ha invece difetti è di vitale importanza per la programmazione della produzione successiva. Conoscere la percentuale di pezzi scartati permetterà all'azienda di mandare in produzione un numero di articoli tali da soddisfare gli ordini dei vari clienti, andando a considerare quelli che sono stati gli scarti nelle produzioni precedenti. Queste informazioni vengono prese da un archivio storico dell'azienda e utilizzate successivamente con il codice precedentemente descritto. Una volta ottenute le regole di associazione bisogna legarle al numero di pezzi da mandare in produzione. L'ingegnere Sara Antonmarioni, dopo aver analizzato e studiato i dati forniti dall'azienda Matrix, ha sviluppato una formula che lega le regole derivanti dal codice con il numero degli articoli da mandare in produzione.

Le informazioni fornite dai dati permettono di ricavare la seguente formula:

$$N = \sum_i n_i * ts_{i-range} * conf_i$$

Con:

- n_i rappresenta le unità dell'articolo i da realizzare (domanda del cliente)
- $ts_{i-range}$ rappresenta il tasso di scarto associato a ciascun range di riferimento per il prodotto
- $conf_i$ rappresenta la confidenza della regola articolo $i \rightarrow \%scarto=ts_{i-range}$

Il risultato sarà il numero di unità da produrre (N) per soddisfare la richiesta del cliente senza rischiare di subire ritardi e conseguenti penali. Il tasso di scarto è stato ricavato con l'ipotesi di produrre 50 unità dell'articolo Apron 36". Sono state considerate le probabilità associate ai vari tassi di scarto, assumendo una distribuzione uniforme dei tassi di scarto all'interno del range di riferimento, alla fine si può affermare che:

- il tasso di scarto associato al range $[0 - 0.167]$ è 0.0835 e si verifica nel 50% dei casi;
- il tasso di scarto associato al range $[0.167 - 0.333]$ è 0.25 e si verifica nel 17% dei casi;
- il tasso di scarto associato al range $[0.167 - 0.500]$ è 0.4165 e si verifica nel 15% dei casi

Quindi se all'azienda Matrix arriva un ordine da parte di un cliente di 50 pezzi, applicando la formula precedentemente descritta, il programmatore della produzione deve inviare alla linea produttiva un ordine di 57 pezzi per far fronte a ritardi e pezzi difettosi per soddisfare la richiesta del cliente.

In conclusione possiamo affermare che il codice pensato per l'azienda Matrix fornisce risultati affidabili e uniti alla formula finale, permette, con le informazioni

che si hanno, di programmare correttamente l'intera produzione nel momento in cui arriva un ordine di un cliente.

Bibliografia

- Long-COVID diagnosis: From diagnostic to advanced AI-driven models
Riccardo Cau, Gavino Faa, Valentina Nardi, Antonella Balestrieri, Josep Puig, Jasjit S Suri, Roberto SanFilippo, Luca Saba, anno 2022
- Using Artificial Intelligence to Detect COVID-19 and Community-acquired Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy
Lin Li, Lixin Qin, Zeguo Xu, Youbing Yin, Xin Wang, anno 2020
- The characteristics and evolution of pulmonary fibrosis in COVID-19 patients as assessed by AI-assisted chest HRCT
Jia-Ni Zou, Liu Sun, Bin-Ru Wang, You Zou, anno 2021
- Artificial intelligence in public services: When and why citizens accept its usage
Sophie Gesk, Michael Leyer, anno 2022
- Improving the remote scheduling of distributed production with process statistics and AI techniques
Alessandra Orsoni, Romeo Bandinelli, anno 2007
- Multi-layered coding-based study on optimization algorithms for automobile production logistics scheduling
Guo Yue, Guo Tailai, Wei Dan, anno 2021
- A predictive maintenance model for optimizing production schedule using deep neural networks
Tiago Zonta, Cristiano André da Costa, Felipe A. Zeiser, anno 2021
- AI based combined scheduling and motion planning in flexible robotic assembly lines
Niki Kousi, Dimosthenis Dimosthenopoulos, Aleksandros-Stereos Matthaiakis, anno 2019
- FR-Tree: A novel rare association rule for big data problem
Mahmoud A. Mahdi, Khalid M. Hosny, Ibrahim Elhenawy, anno 2022
- Mining association rules from COVID-19 related twitter data to discover word patterns, topics and inferences, Paraskevas Koukaras, Christos Tjortjis, Dimitrios Rousidis, anno 2022
- Graph-based modeling using association rule mining to detect influential users in social networks
Tarik Agouti, anno 2022
- On the causation of seafarers' unsafe acts using grounded theory and association rule
He Lan a, Xiaoxue Ma, Weiliang Qiao, Laihao Ma, anno 2022
- A hybrid model for marine accident analysis based on Bayesian Network (BN) and Association Rule Mining (ARM), Emre Özeydi, Remzi Fışkın, Özkan Uğurlu, Jin Wang, anno 2022
- Classification and association rule mining of road collisions for analyzing the fatal severity, a case study, Saba Momeni Kho, Parham Pahlavani, Behnaz Bigdeli, anno 2021
- Multidrug resistance analysis method for pathogens of cow mastitis based on weighted-association rule mining and similarity comparison, Buwen Liang, Xinxing Li, Ziyi Zhang, anno 2021

- Risk prediction and early warning for air traffic controllers' unsafe acts using association rule mining and random forest, Ruihua Xu, Fan Luo, anno 2021
- Comorbidity combinations in schizophrenia inpatients and their associations with service utilization: A medical record-based analysis using association rule mining, Han X1, Jiang, Needleman, Zhou, Yao C, Tang, anno 2021
- Mapping of regulatory actors and processes related to cloud-based energy management environments using the Apriori algorithm
Patrícia Stefan de Carvalho, Julio Cezar Mairesse Siluk, Jones Luís Schaefer, anno 2022
- Inferring implications in semantic maps via the Apriori algorithm, Ni Zeng, Hong Xiao, anno 2020
- Apriori-Based Algorithm for Dubai Road Accident Analysis, Maya John, Hadil Shaiba, anno 2019
- Risk prediction and factors risk analysis based on IFOA-GRNN and apriori algorithms: Application of artificial intelligence in accident prevention, Xie Xuecai, Fu Gui, Xue Yujingyang, anno 2019
- PIN108 Using the Apriori Algorithm to Identify Risk Factors Associated with Survival and Mortality Among COVID-19 Patients, A. Vashisht, C. Holy, S. Shah, N. Elangovanraaj, S. Johnston, P. Coplan, anno 2020
- Apriori and K-Means algorithms of machine learning for spatio-temporal solar generation balancing Nurseda Y. Yürüşen, Bahri Uzunoğlu, Ana P. Talayero, Andrés Llombart Estopiñán, anno 2021
- A heuristic approach for load balancing the FP-growth algorithm on MapReduce
Sikha Bagui, Keerthi Devulapalli, John Coffey, anno 2020
- A guided FP-Growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data, Lior Shabtay, Philippe Fournier-Viger, Rami Yaari, Itai Dattner, anno 2021
- Map-optimize-reduce: CAN tree assisted FP-growth algorithm for clusters based FP mining on Hadoop, J. Ragaventhiran, M.K. Kavithadevi, anno 2020
- FP-Growth based Regular Behaviors Auditing in Electric Management Information System
Jiye Wang, Zhihua Cheng, anno 2018
- Identification of adverse disease agents and risk analysis using frequent pattern mining
Shafiul Alom Ahmed, Bhabesh Nath, anno 2021