Politecnico di Torino

Politecnico di Torino

Corso di Laurea
A.a. 2023/2024
Sessione di Laurea aprile 2024

Progettazione ed integrazione di un modulo di Intelligenza Artificiale all'interno di una piattaforma per la generazione automatica di processi ETL

Relatore:
Tania Cerquitelli

Candidato:
Luigi De Simone

Sommario

Elenco delle figure

ABSTRACT

Con il progredire delle tecnologie e con le nuove necessità delle aziende, gli strumenti di progettazione e gestione delle pipeline ETL (Extract, Transform, Load) si sono evoluti e migliorati col tempo. Tuttavia, il contesto mutevole in cui operano richiede l'uso di lavoro manuale, con conseguenti costi aggiuntivi dovuti ai tempi e agli errori. Con lo sviluppo di algoritmi di IA (Intelligenza Artificiale) generativa sempre più performanti, veloci e accessibili, alcune realtà si stanno adoperando per sfruttare le loro capacità in modo da ridurre gli attuali problemi della gestione dei processi ETL. Nella seguente tesi si propone l'integrazione di un modulo di IA generativa per automatizzare la progettazione di una pipeline ETL. Nella prima parte viene discusso l'attuale stato dell'arte della ricerca negli ambiti ETL e IA generativa, nella seconda parte verranno descritti due prototipi realizzati dal tesista e dai colleghi.

Introduzione

In questo periodo storico le aziende devono prendere decisioni tempestive e informate per mantenere un vantaggio competitivo, per il quale i dati giocano un ruolo essenziale. La raccolta dei dati aziendali e l'implementazione di processi di BI (Business Intelligence) sono diventati strumenti fondamentali per rimanere aggiornati sia sul contesto esterno che su quello interno all'impresa.

Tuttavia, i processi tradizionali di estrazione, trasformazione e caricamento (ETL) che forniscono i dati necessari per i processi di BI sono complessi da gestire e realizzare, e la mole di lavoro manuale coinvolta in questi processi fa aumentare il rischio di errori. Per sopperire a queste difficoltà, molte aziende stanno esplorando soluzioni innovative. Infatti, l'automazione dei processi ETL tramite strumenti avanzati di integrazione dei dati sta diventando sempre più comune. Inoltre, le tecnologie all'avanguardia quali le intelligenze artificiali generative e il machine learning possono semplificare notevolmente la gestione dei dati, in modo da ridurre il lavoro manuale e migliorare l'efficienza complessiva. Queste evoluzioni sono importanti in quanto consentirebbero alle aziende di ottenere informazioni più precise e più rapidamente, favorendo di conseguenza decisioni migliori, oltre a una maggiore reattività alle dinamiche di mercato.

Attualmente, colossi del settore come Google e Oracle si stanno già adoperando per fornire servizi di intelligenza artificiale allo scopo di rendere i loro strumenti di gestione dei dati non solo più accessibili agli utenti meno esperti, ma anche per far risparmiare tempo a quelli più esperti. Tuttavia, è necessario notare che questi sforzi si concentrano principalmente sull'automazione di processi già esistenti, piuttosto che sulla generazione completamente automatica di essi.

L'automazione della generazione di pipeline ETL costituisce una sfida stimolante poiché, nonostante le fasi e le procedure generali siano ben consolidate grazie agli sforzi di ricercatori come Kimball, la parte di dettaglio rimane estremamente soggettiva. Ciò avviene perché la progettazione di tali pipeline è influenzata da variabili esterne, come le richieste

specifiche degli utenti e la natura unica dei loro database e del contesto, e da fattori interni, quali le pratiche e le routine di coloro che le sviluppano.

Proprio per questo motivo l'automazione della progettazione tramite algoritmi statici potrebbe comportare risultati indesiderati. Questi algoritmi potrebbero non essere in grado di catturare in maniera soddisfacente la diversità delle realtà aziendali, delle loro potenzialità e delle loro esigenze. La complessità delle richieste degli utenti e delle loro strutture dati richiedono, quindi, un approccio più sofisticato, che tenga conto di una vasta gamma di possibilità.

Uno di essi potrebbe integrare l'intelligenza artificiale generativa in modo da adattarsi dinamicamente alle specificità di ciascun contesto aziendale. Ciò consentirebbe di superare le limitazioni degli algoritmi statici, fornendo soluzioni più personalizzate e adattabili. In questo modo, l'automazione della generazione di pipeline ETL potrebbe diventare vantaggiosa, riflettendo con precisione la complessità delle richieste da parte delle organizzazioni e i loro cambiamenti.

L'obiettivo della presente tesi è valutare l’opportunità offerta dalla rapida espansione degli algoritmi di intelligenza artificiale generativa. Con le nuove potenzialità di questi modelli e la loro crescente diffusione nel mainstream, si potrebbe realisticamente considerare la possibilità che essi possano apportare un cambiamento significativo anche nell'ambito della gestione e creazione dei processi ETL. Si intravede, infatti, la possibilità di risolvere numerosi problemi legati a questo campo e, probabilmente, sostituire alternative consolidate, come la data virtualization.

La tesi propone di esplorare non solo l'efficacia tecnica degli algoritmi di IA generativa in questo contesto, ma anche la loro applicabilità in contesti reali e la potenziale adozione da parte delle aziende. Si cerca di comprendere se questo approccio innovativo possa non solo offrire vantaggi in termini prestazionali, ma anche superare le limitazioni associate ai metodi convenzionali, aprendo la strada a una nuova era nella gestione dei dati e nei processi ETL.

Nel primo capitolo, è stata condotta una ricerca di diversi articoli accademici che trattano argomenti legati ai temi ETL e intelligenza artificiale generativa. La ricerca è stata effettuata attraverso i siti web IEEEXplore e ScienceDirect, utilizzando parole chiave specifiche quali "etl", "generative ai", "ai" e "generative ai etl". Per fare in modo che gli articoli siano pertinenti con il contesto della tesi, sono stati inclusi solo gli articoli rientranti nell'ambito informatico, poiché il termine "ETL" può avere significati differenti in altri settori. Oltre agli articoli scientifici, sono stati inclusi diversi articoli web allo scopo di avere una visione completa sulle tematiche più recenti che la ricerca accademica non ha ancora affrontato.
Nel secondo capitolo, si fornirà una dettagliata esposizione del funzionamento del progetto sviluppato dal tesista e dai colleghi, approfondendo gli aspetti funzionali e esponendo i vantaggi che il progetto potrebbe offrire rispetto ai metodi tradizionali. Questa sezione mira a fornire una visione esaustiva delle caratteristiche operative del tool, illustrando come esso si differenzi dagli approcci convenzionali e come tali differenze possano tradursi in benefici tangibili per gli utenti.
Il terzo capitolo esporrà lo sviluppo del progetto, con un'analisi dettagliata di tutti i processi implementati e una chiara esposizione del loro funzionamento tecnico. In questa sezione, si esamineranno e verranno giustificate le scelte progettuali adottate, evidenziando le motivazioni dietro le decisioni prese durante il processo di sviluppo. Questo capitolo sarà fondamentale per comprendere il contesto tecnico e le sfide affrontate nell'implementazione del progetto.
Nel quarto capitolo, verranno presentati i risultati di un'operazione di test del progetto, finalizzata a valutare le ipotesi formulate. Si analizzeranno i dati ottenuti durante i test, evidenziando punti di forza e possibili aree di miglioramento. Questa sezione contribuirà a valutare l'efficacia e l'affidabilità del progetto in scenari realistici.
Infine, nel quinto capitolo, saranno esposte le conclusioni del lavoro svolto, con una riflessione sugli obiettivi raggiunti e sulle sfide affrontate. Si esploreranno le potenzialità future relative ai tool, discutendo possibili nuove funzionalità o miglioramenti delle

funzioni esistenti. Sarà quindi delineata la prospettiva a lungo termine del tool e saranno suggerite le possibili vie per lo sviluppo futuro.

1 Attuale stato dell’arte

Nella seguente sezione viene definito il termine “generative ETL” e viene analizzato l’attuale stato della ricerca in tale ambito.

Essendo l’espressione “generative ETL” l’unione dei termini “generative AI” e “ETL”, è utile analizzarli entrambi per avere un quadro chiaro della situazione.

Nel seguente capitolo, il tesista esaminerà approfonditamente il concetto di ETL, illustrandone i benefici, la metodologia di implementazione e gli sviluppi più recenti. Verrà successivamente affrontato il tema delle intelligenze artificiali (IA) generative, analizzando i vari modelli esistenti, valutandone i vantaggi, le applicazioni, i potenziali rischi e delineando le prospettive future di tali tecnologie. Infine, verrà illustrato come l’introduzione delle IA generative nei processi ETL possa risultare vantaggioso per gli utenti.

1.1 ETL

In vari settori economici, la presenza di un data warehouse è essenziale per implementare processi di business intelligence (BI) per ottimizzare le operazioni, migliorare i servizi, gestire in modo efficiente grandi volumi di dati e monitorare le performance dell'azienda. L'obiettivo principale di tali processi è quello di favorire decisioni informate e migliorare l'efficienza complessiva delle attività aziendali.

Tuttavia i database relazionali, tipicamente utilizzati a livello operativo, non sono adeguati alle esigenze di business intelligence a causa di diversi fattori.

La prima causa è attribuibile alla varietà delle fonti dei dati, che rende maggiormente complessa la gestione e l'analisi delle informazioni. Ad esempio, la presenza di standard diversificati può comportare sfide nell'integrazione dei dati. La necessità di convertire o armonizzare formati e strutture diverse può richiedere sforzi considerevoli per garantire la coerenza e l'affidabilità dei dati utilizzati nelle operazioni di business intelligence.

Ulteriore difficoltà è rappresentata dalla struttura dei dati operazionali. Sebbene i database operazionali rispettino le convenzioni di quelli tradizionali per ottimizzare lo spazio, questo risulta in un rallentamento delle interrogazioni dovuto alle operazioni di join tra tabelle.

Ciò è tollerabile per operazioni locali in contesti operativi, ma è limitante per le attività di business intelligence, in quanto è necessario analizzare grandi volumi di dati.
Altro aspetto che compromette l'ottimalità delle sorgenti dati per i processi di business intelligence è la presenza di informazioni non pertinenti o poco significative. Alcuni campi potrebbero risultare superflui o avere una rilevanza limitata nell'ambito dell'analisi BI, contribuendo a creare un surplus di dati non necessari.
Inoltre, un elemento che rende i database operazionali meno adatti per i processi di business intelligence è l'eccessiva granularità dei dati. Un alto livello di dettaglio non apporta alcun beneficio significativo alle attività di BI, ma aumenta considerevolmente il volume complessivo dei dati e i tempi di elaborazione. Questo sovraccarico di dettagli complica l'analisi e l'interpretazione dei dati, mentre le attività di BI si basano su dati aggregati e sintetizzati per ottenere una visione più chiara e gestibile dell'andamento aziendale. Pertanto, è necessaria l'aggregazione dei dati operazionali per ottimizzare il processo di business intelligence.
Anche la mancanza di un elemento temporale in molte fonti di dati è limitante per le operazioni di business intelligence, in quanto l'assenza di tale elemento impedisce la comprensione delle dinamiche aziendali e l'analisi delle tendenze nel tempo.
Per ovviare a questi problemi, da tempo le aziende data-driven ricorrono al trattamento dei dati attraverso processi di ETL in modo da ottenere un modello dati dimensionale, esso è importante per le operazioni di business intelligence in quanto organizza i dati in una maniera tale da effettuare le operazioni e interrogazioni dei dati lungo più dimensioni in maniera più agevole e immediata.

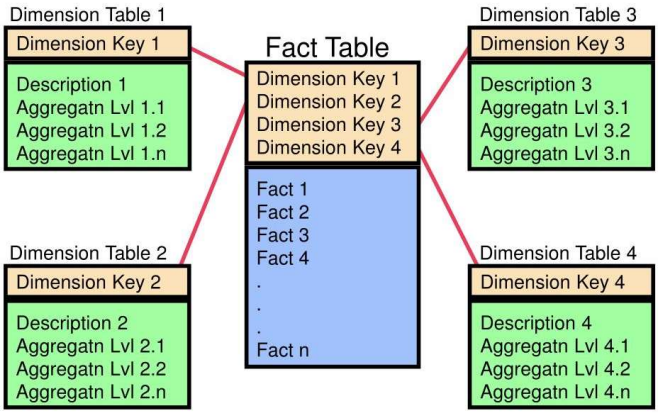

FIGURA 1. SCHEMA A STELLA

Questo modello si realizza nella pratica con delle tabelle disposte secondo uno schema detto “a stella”, che organizza i dati attorno a una tabella centrale detta “tabella dei fatti” con tabelle dette “dimensionali” che la circondano, connesse alla tabella dei fatti attraverso l’uso di chiavi esterne.

Il processo ETL è composto di tre fasi, ognuna dei quali corrisponde a specifiche tabelle. Le fasi corrispondono all'estrazione dei dati da diverse fonti, alla loro trasformazione per adattarli al modello dimensionale e, infine, al caricamento sui database dimensionali.

1.1.1 Vantaggi dei processi ETL

L'implementazione dei processi ETL offre notevoli vantaggi nell'ambito aziendale [25, 26]. In primo luogo, la trasformazione dei dati mira a ridurre i tempi di elaborazione, consentendo un accesso rapido ai dati integrati e trasformati. Questo acceleramento del processo è cruciale per migliorare la tempestività delle decisioni aziendali.

In secondo luogo, il processo di correzione degli errori migliora la qualità dei dati e ha un impatto significativo sulla qualità delle decisioni aziendali. Eliminando dati incoerenti o errati, si promuove l'affidabilità delle informazioni utilizzate per prendere decisioni strategiche.

Infine, l'automazione della migrazione e della trasformazione dei dati riduce la dipendenza dal lavoro umano, diminuendo la probabilità di errori e garantendo una maggiore precisione complessiva nel processo. Ciò porta a una gestione più efficiente e affidabile dei dati aziendali, contribuendo a migliorare l'efficienza operativa e la competitività complessiva dell'organizzazione.

Un campo di applicazione dove le tecnologie ETL risultano vantaggiose è la ricerca sulle malattie, tra le quali l’Alzheimer. Tale processo si confronta con diverse sfide nell'analisi dei dati provenienti da una vasta gamma di fonti, tra cui studi clinici elettronici e dati provenienti da diverse fonti. Questa eterogeneità dei dati spesso richiede competenze tecniche specifiche da parte dei team di ricerca, il che può rallentare significativamente le operazioni. Inoltre, la mancanza di interoperabilità tra dataset con caratteristiche diverse rende difficile l'armonizzazione dei dati e la loro combinazione per analisi più ampie. Per

affrontare queste sfide, viene proposto un approccio innovativo attraverso lo sviluppo di un tool web specializzato nella gestione e progettazione di pipeline ETL con un'interfaccia grafica intuitiva [1], che mira a semplificare il processo di progettazione delle pipeline ETL anche per coloro che non possiedono competenze tecniche avanzate, consentendo così una maggiore partecipazione e collaborazione tra i membri del team di ricerca. Il tool, denominato BIcenter-AD, si articola attraverso diverse fasi, tra cui l'estrazione dei dati dalle fonti disponibili, l'acquisizione dei metadati associati a tali dati, la trasformazione dei dati in un formato comune utilizzando i mappings derivati dall'analisi dei metadati e infine il caricamento dei dati nel database di destinazione. Le caratteristiche chiave di BIcenter includono un editor grafico con funzionalità di drag and drop, che rende la progettazione delle pipeline ETL intuitiva e accessibile. Inoltre, il tool integra funzionalità come Usagi Mapper, che consente l'importazione dei mappings generati da altri strumenti come Usagi, facilitando ulteriormente il processo di armonizzazione dei dati. L'adozione di BIcenter offre diversi vantaggi, tra cui una maggiore intuitività nell'uso dell'editor e delle funzionalità offerte, nonché la capacità di integrare e combinare più dataset in modo efficiente, garantendo al contempo l'interoperabilità tra di essi. Questo approccio innovativo promette di semplificare e accelerare notevolmente il processo di gestione e analisi dei dati, consentendo ai ricercatori di concentrarsi maggiormente sull'analisi e l'interpretazione dei risultati piuttosto che sulle complessità tecniche legate alla manipolazione dei dati.
Generalmente, la creazione del processo ETL e i metodi utilizzati dipendono dal contesto, dal soggetto che li crea e dagli obiettivi. Ai fini della tesi verranno definite le metodologie con cui Mediamente Consulting srl, società presso la quale il tesista ha svolto un tirocinio curriculare, realizza le tre fasi, al fine di rendere comprensibili al lettore le scelte adottate durante la creazione del prototipo.

1.1.2 Fasi dell’ETL

Il processo di ETL si divide nei tre livelli L0, L1 e L2.

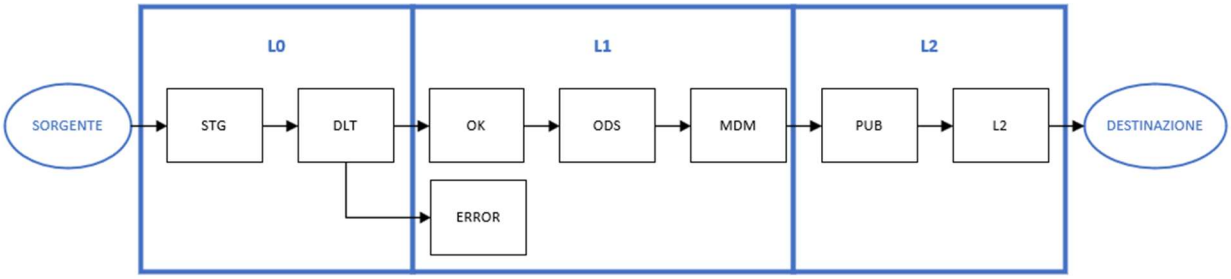

FIGURA 2. SCHEMA RAPPRESENTANTE IL PROCESSO ETL E DEI TRE LIVELLI

1.1.2.1 L0

La fase iniziale implica l'acquisizione dei dati dal sistema sorgente, divisa in due sottoprocessi: staging e cattura del delta, gestiti attraverso le tabelle STG e DLT.

Lo staging coinvolge il trasferimento completo dei dati dalla tabella sorgente alla tabella STG, accompagnato dall'introduzione di una nuova variabile chiamata JOBID. Questo identificatore numerico rappresenta singolarmente ogni processo, con le sue cifre che indicano la data e l'ora di esecuzione nel formato *YYYYMMGGhhmmss*. Tale procedura è cruciale poiché nelle fasi successive sarà necessario partizionare i dati di STG in base ai vari caricamenti.

Nella fase di cattura del delta, l'obiettivo è caricare solo le modifiche rispetto all'ultimo processo dal sistema sorgente, ottimizzando così le prestazioni quando si lavora con un volume considerevole di dati. Se la tabella sorgente contiene un campo che indica la data di caricamento, il confronto tra le date consente di identificare i cambiamenti, memorizzando l'ultima data letta in una tabella apposita. Altrimenti, si eseguono due query SQL MINUS sulle ultime due partizioni caricate su STG:

- La prima operazione confronta dalla partizione più recente a quella meno recente per individuare i record aggiunti;
- La seconda operazione confronta dalla partizione meno recente a quella più recente per individuare i record cancellati.

Inoltre, in questa fase viene aggiunto il campo FLG_NEG, che assume il valore 0 per i record aggiunti e 1 per quelli cancellati. I record risultanti vengono poi caricati nella tabella DLT.

1.1.2.2 L1

Nella fase successiva del processo, si effettua una selezione e una conservazione dei dati. Durante la selezione, viene verificata l'integrità dei dati, con particolare attenzione alla correttezza. Qualsiasi dato che non sia conforme viene identificato e inserito in una specifica tabella per un'ulteriore analisi e, se possibile, correzione e successivo riutilizzo. Gli errori rilevati includono valori inesatti, dati mancanti, duplicati e record con riferimenti esterni non validi (integrità referenziale).

I dati che superano con successo i controlli vengono quindi caricati nella tabella "OK". È importante notare che questa tabella deve essere svuotata ad ogni esecuzione per garantire la coerenza dei dati.

Successivamente, i dati devono essere aggiornati e conservati se sono stati soggetti a modifiche. Per realizzare questo, si utilizza un comando di fusione (MERGE) tra la tabella ODS (Operational Data Storage) e la tabella "OK". Durante questo processo, vengono anche aggiunti attributi come JOBID e timestamp di aggiornamento per tracciare le modifiche.

Nel caso in cui i dati provengano da fonti diverse, è necessario integrarli in una singola tabella principale, conosciuta come MDM (Master Data Management). In questa tabella, viene introdotta una chiave surrogata (sk) per ottimizzare le operazioni di unione dei dati.

1.1.2.3 L2

Infine, nella fase conclusiva del processo, si eseguono gli ultimi aggiustamenti in base al sistema di visualizzazione prescelto e si procede con il caricamento finale dei dati nella tabella di destinazione.

Nella tabella PUB, si assicura che le chiavi surrogate presenti nella tabella dei fatti corrispondano a quelle delle tabelle dimensionali. Successivamente, i dati vengono trasferiti nella tabella OUT, che rappresenta la tabella finale contenente i dati pronti per essere utilizzati nelle attività di analisi dei dati.

Infine, viene eseguita un'operazione di aggregazione nella tabella OUT per ottimizzare i calcoli e velocizzare il processo di analisi.

1.1.3 Stato attuale della ricerca

Da quando i primi processi ETL sono stati implementati e fino ad oggi, nuove tecnologie e nuove sfide hanno determinato cambiamenti importanti nel modo in cui i processi ETL vengono progettati ed eseguiti.

1.1.3.1 Big Data

Il primo di essi è la nascita del contesto big data, caratterizzato da un volume esponenzialmente maggiore di dati sempre più diversi tra loro, per cui si è rivelato necessario avere processi di data warehousing più veloci, capienti e capaci di elaborare dati semi-strutturati e non strutturati quali video, immagini e testi. Una prima soluzione a questo problema è il cloud computing, che consentirebbe spazio e potenza di calcolo potenzialmente infiniti.

Diversi articoli accademici rivelano le nuove necessità da affrontare nel nuovo ambiente big data.

In una review del 2018 vengono individuate diverse potenziali soluzioni che utilizzano la tecnologia cloud [2]. Stando ad essa, oltre alla pianificazione dei processi ETL per i dati memorizzati, è necessario catturare e processare le data streams, ossia i flussi di dati continui, che richiedono l'estrazione istantanea di essi. In questo scenario, diventa cruciale non solo l'utilizzo di query eseguite in modo continuo, ma anche la sincronizzazione tra le attività di aggiornamento dei dati e le estrazioni per evitare conflitti nelle query. L'articolo sottolinea la necessità di affrontare il problema della "query contention" durante le esecuzioni in tempo reale. La contention può verificarsi quando più processi tentano di accedere o modificare i dati contemporaneamente, causando ritardi e inefficienze nell'elaborazione delle query. Inoltre, l'analisi delle tecniche di ETL in tempo reale rivela che, sebbene le performance attuali siano adeguate per la cattura immediata dei dati strutturati, si identifica un divario nella gestione dei dati non strutturati. Questo suggerisce

che le attuali soluzioni ETL in tempo reale potrebbero non essere ottimali per affrontare completamente la complessità dei dati non strutturati.
in un altro articolo si sono valutate le nuove sfide dei sistemi ETL dal punto di vista della velocità di elaborazione e di esecuzione dei processi [3]. L'articolo afferma che affrontare l'implementazione di sistemi ETL in un ambiente near real-time presenta una serie di sfide che vanno ben oltre le complessità di un contesto tradizionale. In primo luogo, la stima dei costi e delle risorse necessarie può risultare complessa, dato che dipende da fattori quali la complessità dei dati, il loro volume, la frequenza di aggiornamento e gli obiettivi di latenza. Queste stime vanno costantemente aggiornate per adeguarsi a eventuali cambiamenti nei requisiti o nell'ambiente circostante. Le caratteristiche specifiche dell'ambiente near real-time richiedono un sistema ETL capace di operare a velocità elevate e con una bassa latenza mantenendo, allo stesso tempo, un'elevata disponibilità allo scopo di assicurarsi che i dati siano sempre accessibili per l'analisi. Affrontare l'estrazione dei dati comporta il rischio di sovraccaricare i sistemi di origine a causa della frequenza elevata di acquisizione. Inoltre, la necessità di aggiornamenti continui richiede un'attenzione costante agli sviluppi in corso nei sistemi di origine. Le trasformazioni devono essere eseguite in tempi più brevi, portando a sfide legate all'efficienza e alla complessità delle stesse in un contesto near real-time. Per quanto riguarda il caricamento, è cruciale ottimizzare le prestazioni OLAP per evitare sovrapposizioni con il processo di caricamento dei dati. Il caricamento in tempo reale diventa un requisito essenziale per garantire la costante disponibilità dei dati per l'analisi. Oltre a queste sfide, è vitale considerare anche aspetti come la sicurezza dei dati, poiché i dati in tempo reale sono più vulnerabili alle minacce alla sicurezza. La governance dei dati diventa altrettanto essenziale per assicurare l'accuratezza e l'affidabilità dei dati in tempo reale. Il monitoraggio continuo delle prestazioni e la manutenzione sono inoltre parte integrante del processo, garantendo l'efficienza e l'efficacia del sistema ETL. Nonostante le complessità e gli investimenti significativi richiesti, l'implementazione di sistemi ETL in un ambiente near real-time offre vantaggi cruciali, come una migliore visibilità aziendale,

maggiore agilità nel rispondere ai cambiamenti del mercato e miglior efficienza operativa, contribuendo a prendere decisioni aziendali più informate.

In un'analisi dello stato dell'arte [24] delle modalità di implementazione di ETL in cloud per gestire la diversità dei dati, emergono diverse soluzioni proposte. Tra queste, l'utilizzo di ontologie si presenta come una strategia interessante, consentendo la gestione di dati strutturati e semistrutturati, pur tralasciando quelli non strutturati. Tuttavia, si evidenzia la possibile perdita di informazioni dovuta alla rappresentazione unificata dei dati, con conseguente compromissione dell'accuratezza. Un approccio alternativo consiste nell'impiego di tecnologie di semantic enhancement, che permettono la creazione di un dataspace in cui tutti i dati possono essere rappresentati, organizzati e interpretati in modo coerente, indipendentemente dalla loro origine o struttura originale. Tuttavia, va notato che questo modello è statico e richiede la generazione di un nuovo dataspace ad ogni nuova struttura dei dati, comportando una riduzione della velocità operativa. Un'altra prospettiva si basa sull'utilizzo di modelli semantici progettati attraverso ontologie, tuttavia, è importante sottolineare che spesso mancano risultati sperimentali a sostegno di questa metodologia. Per quanto riguarda l'estrazione di concetti principali da file multimediali, una modalità flessibile è stata identificata, in grado di trattare autonomamente le sorgenti senza integrarle in un'unica fonte. Tuttavia, è necessario evidenziare la limitazione di questo processo a testi, immagini e video. In conclusione, emerge che la maggior parte degli studi attuali fa uso di tecnologie basate su sintassi, e la questione della dinamicità dei tipi di dato nel tempo non è ancora stata completamente risolta. Inoltre, si nota una carenza di attenzione verso l'analisi dei costi associati all'utilizzo delle soluzioni basate su cloud, suggerendo la necessità di approfondimenti in questo ambito per una valutazione completa delle strategie implementative proposte.

Esempio di adattamento dei processi ETL al nuovo contesto Big Data è l’ideazione di un framework di gestione dei metadati da parte di Suleykin e Panfilov [4]. Il contesto tradizionale di ETL si è dimostrato efficace nel processare dati in modalità periodica e a pacchetti. Tuttavia, con l'avvento del contesto Big Data, in cui i dati vengono emessi in

modo continuo, si rende necessario un approccio più flessibile e dinamico. A tale scopo, un articolo propone l'implementazione di un sistema ETL basato su metadati, integrato con Apache Airflow, al fine di automatizzare le operazioni tipiche dei processi ETL. Il framework di gestione dei metadati offre un unico punto di accesso per la gestione completa dei metadati, compresa la creazione dei Directed Acyclic Graphs (DAG) di Airflow, che costituiscono le pipeline di dati. Tra le funzionalità principali del framework vi sono un'interfaccia di gestione dei metadati, la costruzione automatica delle pipeline di caricamento, l'automazione dell'implementazione di nuove entità, e l'adozione di un'architettura coesa in linea con le convenzioni di denominazione. Il sistema ETL è strutturato attorno ai DAG di Airflow, con gli script eseguiti che supportano diversi linguaggi (shell, Python, PostgreSQL, Hadoop, ecc.). Il cuore del sistema è rappresentato da un framework di metadati, ospitato su un database PostgreSQL. Questo framework contiene informazioni cruciali sulla struttura del sistema quali le sorgenti di dati e la struttura delle tabelle sorgenti, e la configurazione del serving layer. Include anche dettagli sui modelli di dati versionati, la sincronizzazione degli ambienti di sviluppo e l'evitamento di hard coding e file di configurazione extra. Il processo ETL si articola attraverso fasi ben definite, tra cui la registrazione dei file in arrivo su directory HDFS nella fase di staging (STG), la trasformazione dei file XML nel formato di output richiesto nella fase di exchange (EXCH), l'aggregazione dei dati nel batch view (BV), e la scrittura sul database PostgreSQL (DDS_STG). Ulteriori attività comprendono la creazione dell'archivio Hadoop Archive (HAR), la pulizia di EXCH, e la creazione dei data mart attraverso lo script PG_etl_dm. La gestione delle entità senza corrispondenze o collegamenti noti è parte integrante del processo, garantendo la conformità all'architettura di sistema e alle convenzioni di denominazione. Infine, il sistema adotta un'interfaccia grafica per semplificare l'interazione e la gestione delle operazioni, fornendo una visione chiara delle fasi coinvolte nel processo ETL.

1.1.3.2 Agile e DevOps

Con il tempo, la gestione del lavoro si è dovuta trasformare come meccanismo di adattamento ai cambiamenti del mercato. Da ciò, sono risultate nuove metodologie di lavoro che favoriscono la reattività delle organizzazioni alla mutevolezza del mondo esterno.

Il primo di essi è la filosofia agile, diventata sempre più diffusa nel contesto informatico. Queste metodologie sono progettate per favorire la flessibilità, la reattività agli imprevisti e lo sviluppo incrementale. Le aziende che si ispirano a tale filosofia fanno uso dei due metodi kanban e scrum.

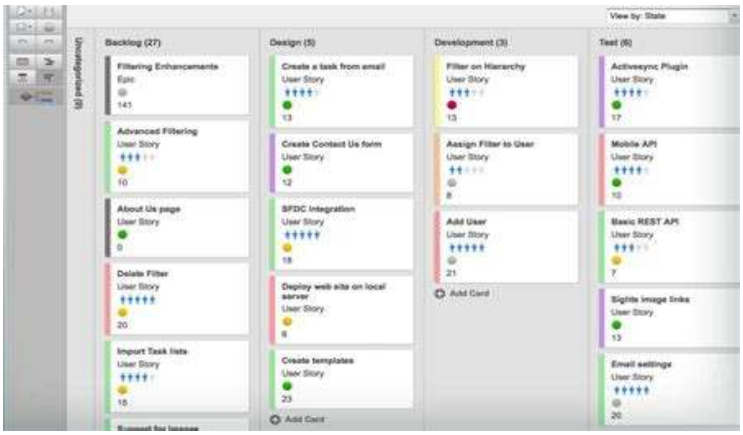

FIGURA 3. ESEMPIO DI KANBAN BOARD

Il metodo Kanban, derivato dai principi della produzione snella, offre un approccio visuale e intuitivo alla gestione del lavoro. Utilizzando una lavagna (reale o virtuale) e i kanban, che rappresentano i singoli processi da svolgere, il team può monitorare facilmente lo stato di avanzamento del lavoro. Questo metodo consente di visualizzare chiaramente quali attività sono in corso, quali sono state completate e quali sono in coda per essere affrontate. La trasparenza e la chiarezza visiva offerte da Kanban facilitano la pianificazione e l'organizzazione del lavoro.

Dall'altro lato, lo Scrum è un framework iterativo che suddivide il progetto in unità di tempo definite come sprint. Gli sprint sono periodi di tempo fissi, tipicamente di 2-4 settimane, durante i quali il team si impegna a consegnare un incremento del prodotto. La struttura di Scrum favorisce una progressione graduale e continua, consentendo al team di adattarsi rapidamente ai cambiamenti e di fornire valore al cliente ad ogni iterazione. Inoltre, la comunicazione continua e la collaborazione stretta con il cliente durante tutto il processo

contribuiscono a garantire che il prodotto finale soddisfi appieno i requisiti e le aspettative del cliente.

Inoltre, le aziende nel settore informatico hanno subito una trasformazione significativa attraverso l'adozione di pratiche volte a favorire la collaborazione continua tra i diversi dipartimenti. Un esempio recente di questa evoluzione è l'implementazione di procedure DevOps, che si focalizzano sull'integrazione e la collaborazione costante tra i team di sviluppo (Dev), operazioni (Ops) e controllo qualità e sicurezza.

L'adozione di metodi DevOps ha portato diversi benefici per le aziende. Innanzitutto, si è registrato un miglioramento generale della qualità dei software rilasciati. La collaborazione continua tra i team permette di individuare e risolvere tempestivamente eventuali problemi, garantendo una maggiore affidabilità dei prodotti finali. Questo, a sua volta, ha contribuito a incrementare la soddisfazione del cliente, poiché i software sono più stabili e rispondono meglio alle aspettative degli utenti.

Inoltre, l'implementazione di DevOps ha portato a una riduzione significativa dei tempi di sviluppo. La collaborazione continua e l'automazione di processi consentono un flusso più rapido dalla fase di sviluppo a quella di produzione. Ciò favorisce la reattività al mercato e alla concorrenza, consentendo alle aziende di adattarsi rapidamente alle richieste e alle evoluzioni del settore.

Il recente sviluppo di metodologie di lavoro agili e DevOps non solo ha reso più flessibile lo sviluppo di processi ETL, ma ha anche ipotizzato lo sviluppo di soluzioni ETL basate sulla filosofia agile.

In un articolo si è proposto un tool di generazione di pipeline ETL, chiamato Agile ETL, che sia low cost e accessibile agli utenti meno tecnici senza dover sacrificare le performance [5]. Agile ETL rappresenta un sofisticato strumento progettato per semplificare e automatizzare integralmente il complesso processo di Extract, Transform e Load (ETL). L'applicazione offre un approccio completo, partendo dall'estrazione dei dati da diverse fonti, come database e server web, per poi procedere con la loro trasformazione e successivo caricamento nel data warehouse. Uno degli elementi distintivi di Agile ETL è la sua capacità

di garantire un monitoraggio continuo durante l'esecuzione dei processi ETL. Questo significa che il software è in grado di rilevare immediatamente eventuali errori o anomalie, notificando gli utenti in modo tempestivo per consentire interventi immediati e garantire la continuità delle operazioni. L'interfaccia utente di Agile ETL è stata progettata con l'obiettivo di essere intuitiva e accessibile a un vasto pubblico, anche a coloro che non hanno una vasta esperienza nel campo dell'ETL. Questa caratteristica si traduce in un'interazione più agevole con il software, eliminando barriere per gli utenti meno esperti. Un aspetto cruciale di agile ETL è il suo impegno per mantenere un costo contenuto, rendendo l'automazione dei processi ETL accessibile a un'ampia gamma di aziende e organizzazioni, indipendentemente dalle loro dimensioni. Le componenti principali di Agile ETL includono l'Estrattore, responsabile di recuperare i dati da diverse fonti e trasferirli in un'area di staging. Successivamente, il Trasformatore entra in gioco, applicando le trasformazioni necessarie ai dati nella fase di staging. Infine, l'Integratore sposta i dati trasformati dallo staging al data warehouse, completando il processo di ETL. Il software include anche un Indicator Model che offre una panoramica dettagliata delle prestazioni dei processi ETL. La funzionalità Scope consente di limitare la portata dell'elaborazione, concentrandosi su record specifici. La Tabella Centralizzata degli Eventi registra con precisione tutte le attività durante l'esecuzione dei processi ETL. L'Interfaccia Utente Web aggiunge un livello di flessibilità, consentendo la gestione e il monitoraggio dei processi ETL da qualsiasi dispositivo connesso a Internet. In confronto ad altri strumenti ETL disponibili, Agile ETL si distingue per le sue solide prestazioni e l'affidabilità nelle operazioni di estrazione e integrazione dati, rendendolo una soluzione completa e affidabile per l'automazione e il monitoraggio dei processi ETL.

1.1.3.3 Open source e low code

Una trasformazione significativa dei tools ETL è dovuta alla transizione dal modello closed source, caratterizzato da una forte complessità tecnica e da requisiti di conoscenza degli applicativi avanzati, al modello open source, che promuove l’usabilità, la personalizzazione e l’accessibilità in termini di costi agli utenti.

Il concetto di open source si riferisce al software il cui codice sorgente è reso disponibile al pubblico, consentendo agli utenti di utilizzarlo, modificarlo e distribuirlo liberamente. Questa filosofia ha introdotto diversi vantaggi che hanno contribuito alla sua crescente popolarità e diffusione ampia.

La trasparenza del codice sorgente è un pilastro fondamentale, contribuendo a garantire una maggiore sicurezza informatica. La possibilità per gli utenti di esaminare e comprendere il funzionamento interno del software favorisce un approccio collaborativo alla rilevazione e risoluzione di vulnerabilità, aumentando la robustezza complessiva del sistema.

L'aspetto personalizzabile del codice rappresenta un altro aspetto chiave. La flessibilità offerta dalla capacità di modificare e adattare il software alle specifiche esigenze del contesto e degli utenti consente una maggiore aderenza alle particolari richieste e requisiti, differenziandosi così dalle soluzioni rigide e standardizzate.

Dal punto di vista economico, l'adozione di software open source può presentare notevoli vantaggi in termini di costi. In genere, l'accesso gratuito al codice sorgente e la possibilità di utilizzarlo senza oneri di licenza si traducono in una riduzione dei costi iniziali e operativi, contribuendo a rendere il software più accessibile a una vasta gamma di utenti.

La presenza di comunità di sviluppatori online rappresenta un elemento dinamico che caratterizza il panorama open source. La collaborazione tra sviluppatori provenienti da diverse parti del mondo può portare a una rapida identificazione e risoluzione di problemi, contribuendo a mantenere il software aggiornato e affidabile. Questo contrasta con l'approccio tradizionale dei software proprietari, che spesso richiedono il supporto del servizio clienti dell'azienda sviluppatrice.

Inoltre, la continuità dello sviluppo da parte di una comunità di sviluppatori diversificata rappresenta un notevole vantaggio. La possibilità che altri sviluppatori prendano in carico il progetto garantisce la longevità e l'evoluzione del software anche in assenza dell'apporto iniziale del creatore originale.

Infine, l'adozione di soluzioni open source contribuisce a evitare dipendenze e vincoli nei confronti di singole aziende. Questa libertà da lock-in consente agli utenti di mantenere un maggiore controllo sulla propria infrastruttura tecnologica, mitigando i rischi associati alle dinamiche del mercato e alle evoluzioni aziendali.

Un altro cambiamento significativo degli strumenti ETL è attraverso l'adozione dell'approccio low code nello sviluppo dei nuovi tool. Il termine "low code" si riferisce a una metodologia nello sviluppo del software che mira a semplificare il processo di creazione di applicazioni riducendo la quantità di codice tradizionale che gli sviluppatori devono scrivere. In altre parole, con la tecnologia low code, è possibile sviluppare applicazioni utilizzando un numero inferiore di linee di codice rispetto agli approcci tradizionali. Ciò consente di avere software semplificati, che sono vantaggiosi in quanto favoriscono l'accessibilità per i non sviluppatori, hanno costi e tempi di sviluppo ridotti, facilitano la manutenzione e l'integrazione con altri software e rendono più facile lo sviluppo, con il beneficio aggiunto di una riduzione degli errori. Un esempio di applicativo low code e open source in ambito ETL è Apache NiFi.

Uno studio propone un processo ETL realizzato tramite NiFi che potrebbe avere dei risvolti significativi nell'ambito Industria 4.0 e nello sviluppo delle Smart City [6], nei quali l'integrazione e l'elaborazione dei dati spaziali sono cruciali, per cui viene coinvolto l'uso di NiFi in quanto facilita la gestione fluida dei flussi di dati. Una funzionalità del processo è quella di caricare e uniformare dati provenienti da varie fonti, come sensori IoT e social network, sfruttando la flessibilità offerta da NiFi. Un aspetto distintivo di NiFi è la sua interfaccia grafica intuitiva, che semplifica notevolmente la configurazione e il monitoraggio dei flussi di dati. Questa caratteristica non solo accelera il processo di sviluppo, ma rende anche l'interazione con il sistema più accessibile anche per coloro che non sono esperti tecnici. Inoltre, la gestione delle autorizzazioni e dei permessi in NiFi offre un controllo granulare sull'accesso ai dati, garantendo la sicurezza e la conformità normativa. Questo è fondamentale quando si trattano informazioni spaziali sensibili in un contesto urbano intelligente. La capacità di fornire supporto tecnico è un altro vantaggio

significativo offerto da NiFi. Questo assicura che eventuali problemi siano affrontati prontamente, mantenendo la continuità operativa e ottimizzando le prestazioni del sistema. Un punto di forza del progetto è la possibilità di facilitare la comunicazione tra sistemi NiFi diversi. Questo favorisce l'interoperabilità e la collaborazione tra varie infrastrutture di smart city, consentendo una gestione unificata dei dati spaziali. In sintesi, l'impiego di Apache NiFi per il processamento e l'integrazione dei dati spaziali nell'ambito delle smart cities si traduce in un approccio avanzato e versatile. La sua interfaccia user-friendly, il controllo delle autorizzazioni, il supporto tecnico e la facilitazione della comunicazione tra sistemi contribuiscono a rendere questo strumento una scelta strategica per sostenere lo sviluppo e la gestione efficiente delle città intelligenti.

In un altro articolo viene fatta una descrizione generale del tool Talend e confrontato con un altro tool commerciale Informatica [7]. Talend viene descritta come una suite di strumenti ETL che offre sia soluzioni commerciali che open source. Tra queste due opzioni, Talend è particolarmente rinomato e ampiamente utilizzato nella community. Ciò è in gran parte dovuto alla sua popolarità e alla sua flessibilità. Un aspetto distintivo di Talend è la sua natura intuitiva, rendendo l'esperienza degli utenti più accessibile. Inoltre, il fatto che Talend sia basato su Java contribuisce alla sua robustezza e alla sua capacità di gestire processi complessi di integrazione dei dati. Un punto forte di Talend è la possibilità di scrivere codice personalizzato, consentendo agli sviluppatori di adattare le soluzioni in base alle esigenze specifiche del progetto. Questa caratteristica offre un livello di flessibilità che può essere fondamentale in scenari complessi di integrazione dei dati. Un altro vantaggio significativo di Talend è la sua facilità di distribuzione (deployment). Il processo di implementazione è progettato per essere user-friendly, semplificando la messa in produzione delle soluzioni integrate. Ciò contribuisce a ridurre il tempo e gli sforzi necessari per implementare e gestire con successo progetti di integrazione dei dati utilizzando Talend. Va, tuttavia, tenuto in considerazione che lo studio è del 2021, in quanto Talend è stata acquistata da Qlik nel 2024 e il software non è più open source.

1.1.3.4 Ultimi trend

La virtualizzazione dei dati rappresenta un approccio innovativo all'integrazione dei dati, che si differenzia dalla tradizionale infrastruttura di data warehousing [27, 28]. In questo processo, l'accesso e la manipolazione dei dati avvengono attraverso un software intermedio posizionato tra le sorgenti di dati e l'interfaccia utente. Un elemento chiave di questo approccio è che l'accesso ai dati avviene direttamente a livello delle sorgenti, eliminando la necessità di un data warehouse dedicato. Questo, a sua volta, si traduce in notevoli risparmi nei costi di installazione e gestione dell'infrastruttura.

Un vantaggio significativo della virtualizzazione dei dati è la possibilità di accedere ai dati operativi più recenti, consentendo alle aziende di prendere decisioni informate basate sui dati più aggiornati. Questa caratteristica può essere particolarmente vantaggiosa per le aziende più piccole che gestiscono un volume limitato di dati.

Tuttavia, è importante considerare che la virtualizzazione dei dati potrebbe incontrare delle limitazioni, soprattutto in presenza di dataset di dimensioni significative. Al di là di una certa dimensione, l'integrazione attraverso un data warehouse potrebbe diventare preferibile. La gestione efficiente di grandi quantità di dati potrebbe essere più complessa senza l'infrastruttura consolidata fornita da un data warehouse.

In sintesi, la scelta tra virtualizzazione dei dati e data warehousing dipende dalle esigenze specifiche dell'azienda, dalle dimensioni del dataset e dalle risorse disponibili. Mentre la virtualizzazione dei dati offre flessibilità e risparmi sui costi per aziende più piccole, il data warehousing potrebbe essere la scelta preferita quando si gestiscono volumi di dati considerevoli.

In un articolo dedicato al miglioramento della qualità dei dati attraverso l’utilizzo della data virtualization [8], si propone l'impiego di questa tecnologia per migliorare la qualità dei dati in ingresso. L'analisi dei risultati sottolinea diversi vantaggi rispetto ai tradizionali processi ETL. La caratteristica di accesso diretto ai dati operazionali, senza la necessità di movimentare i dati, si traduce in un significativo risparmio di tempo durante l'elaborazione. Inoltre, l'autore afferma che le operazioni sui dati diventano più semplici e riutilizzabili in

contesti diversi, grazie alla possibilità di trasformare i dati da un formato all'altro senza complicazioni. La piattaforma supporta, inoltre, operazioni come il "clustered deployment" e le "federated query", ottimizzando i tempi e promuovendo la scalabilità.

Un'analisi comparativa tra Data Warehousing (DW) e Data Virtualization (DV) [9] evidenzia distinzioni chiave nell'approccio all'architettura dati e all'integrazione. In termini di modelli di dati, il DW si basa su strutture fisiche, come tabelle di fatti e dimensioni separate, mentre il DV adotta modelli virtuali, aggregando informazioni da sorgenti diverse senza duplicarle. Relativamente alle connessioni ai dati, il DW non è direttamente collegato alle sorgenti originali, mentre il DV mantiene una connessione diretta, agevolando l'accesso in tempo reale. L'aggiornamento dei dati segue un calendario periodico nel DW (giornaliero, settimanale, mensile), mentre il DV offre dati "on-the-fly" e "on-demand," garantendo maggiore tempestività. L'accesso in tempo reale è problematico nel DW, ma agevolato nel DV, consentendo analisi più immediate. Riguardo a modifiche e scalabilità, il DW richiede un redesign complesso per nuove sorgenti, mentre il DV si adatta agevolmente, garantendo maggiore scalabilità. L'isolamento degli strumenti BI è limitato nel DW, mentre il DV fornisce un livello di astrazione che semplifica l'utilizzo. Le funzionalità avanzate differiscono, con il DW che offre data mart fisici e maggiore complessità di modellazione, mentre il DV fornisce data mart virtuali e una modellazione semplificata. La disponibilità dei dati nel DW non richiede necessariamente che siano online, mentre il DV impone la presenza online delle sorgenti per garantire l'accesso in tempo reale. Il DW si concentra sull'archiviazione centralizzata di dati storici, mentre il DV segue un approccio "on-demand," offrendo un ambiente virtuale per l'accesso a informazioni integrate da diverse sorgenti. La flessibilità risulta limitata nel DW, adatto a scenari consolidati con dati storici stabili, mentre il DV si dimostra più flessibile per contesti dinamici che richiedono agilità e accesso in tempo reale. Ulteriori considerazioni includono la mancanza di specifiche sulla natura dei dati e l'influenza di fattori economici e di governance nella scelta tra DW e DV. L'evoluzione tecnologica ha anche introdotto soluzioni ibride che combinano aspetti di entrambi gli approcci. In conclusione, la decisione tra DW e DV richiede un'analisi

dettagliata delle esigenze specifiche e degli obiettivi aziendali. Queste differenze sottolineano come la scelta tra Data Warehouse e Data Virtualization debba tener conto delle esigenze specifiche del contesto e delle preferenze in termini di flessibilità, complessità e connettività con i sorgenti dati.

1.2 Generative AI

Le generative AI sono modelli di machine-learning unsupervised capaci di generare elementi multimediali e testi in base a un prompt dato dall'utente. Sono solitamente utilizzati sotto forma di chatbot, per generare testi e codice, e per generare immagini.

1.2.1 Funzionamento delle IA generative

Nonostante i modelli di intelligenza artificiale generativa abbiano recentemente guadagnato grande popolarità, è importante notare che versioni rudimentali esistevano già dai primi anni del 1900. Uno dei modelli più noti è il modello markoviano, ideato dal matematico russo Andrei Markov nel 1906. Questo modello è caratterizzato dalla proprietà markoviana [28], che stabilisce che la probabilità di uno stato futuro dipende solo dallo stato attuale e non da tutti gli stati passati del processo. I modelli markoviani sono stati storicamente impiegati nel suggerimento automatico delle parole, ad esempio sulla tastiera di uno smartphone, basandosi sull'ultima scritta. Tuttavia, a causa della loro semplicità e della limitazione imposta dalla proprietà markoviana, questi modelli risultano efficaci solo per processi relativamente semplici.

La crescente popolarità dei modelli di intelligenza artificiale generativa è stata favorita da due importanti sviluppi [29]. In primo luogo, l'ampio contesto di big data ha consentito l'utilizzo di dataset sempre più vasti durante la fase di addestramento dei modelli di intelligenza artificiale. In secondo

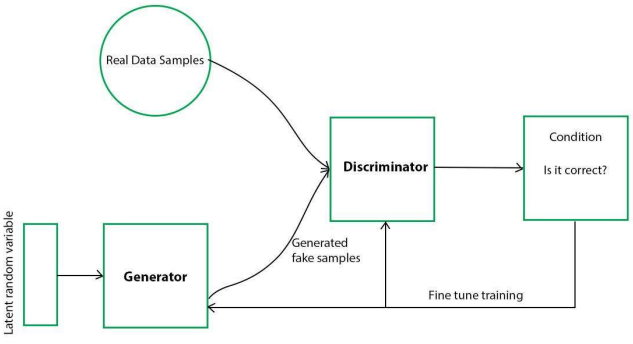

FIGURA 4. FUNZIONAMENTO DELLA GAN

luogo, gli avanzamenti nella ricerca hanno portato alla creazione di modelli più complessi ma anche più performanti.

Uno dei principali contributi a questa evoluzione è rappresentato dalle reti generative avversarie (GAN) [31,32,34]. Questo approccio coinvolge due modelli distinti: il modello generativo si impegna a creare dati che assomigliano il più possibile al dataset di apprendimento, mentre il modello discriminativo ha il compito di distinguere i dati reali da quelli generati. Il risultato del test viene quindi comunicato al modello generativo, e questo ciclo si ripete. Grazie a questo processo iterativo, il modello generativo diventa sempre più abile nel generare dati che sono difficilmente distinguibili da quelli del mondo reale.

Attualmente, il modello di GAN più diffuso e potente in circolazione è StyleGAN. Progetto open source sviluppato da Nvidia, StyleGAN è un modello capace di generare immagini in alta risoluzione, ed è tanto potente da poter generare immagini all'occhio umano indistinguibili rispetto alle immagini della vita reale.

Le GAN sono note per la loro capacità di generare nuovi dati simili a quelli con cui vengono addestrati. Tuttavia, mostrano una limitata flessibilità quando si tratta di introdurre specifiche variazioni nei dati. Per superare questa limitazione, è stata sviluppata una nuova categoria di modelli denominata autoencoder variazionali, conosciuti anche come VAE [34].

Questi modelli sono costituiti da due reti neurali, comunemente chiamate encoder e decoder. L'encoder si occupa di mappare i dati di input in uno spazio latente caratterizzato da una distribuzione di probabilità nota. Dall'altra parte, il decoder elabora l'output dell'encoder eseguendo il processo inverso, generando una ricostruzione che segue la stessa distribuzione di probabilità dei dati di input.

Per potenziare le performance degli autoencoder variazionali (VAE), spesso viene applicato il principio delle Generative Adversarial Networks (GAN). In questo approccio, il VAE viene addestrato in collaborazione con un discriminatore. Il compito del discriminatore va oltre

la semplice distinzione tra dati reali e quelli generati dal VAE; gli viene anche assegnato il compito di individuare la funzione di similarità tra i dati.
Questa combinazione di approcci consente al VAE di beneficiare delle caratteristiche delle GAN, ottenendo una maggiore qualità nella generazione dei dati e una migliore capacità di adattarsi alle variazioni nei dati di input. L'uso congiunto di VAE e discriminatore rappresenta un passo avanti nell'integrazione di principi di apprendimento generativo e probabilistico per ottenere risultati più avanzati e flessibili.
Un utilizzo delle VAE-GAN è riscontrato nell'ambito della predizione del Remaining Useful Life (RUL), ovvero la durata residua, di un motore aeronautico [10]. Questo modello combina due principi chiave per ottenere stime più accurate. Il metodo si basa sulla riduzione della dimensionalità attraverso l'impiego di un VAE, finalizzato a estrarre le caratteristiche più significative dai dati del motore. Successivamente, il decoder del VAE funge da generatore in una GAN, producendo curve di Health Index (HI) realistiche, rappresentative della salute del motore nel tempo. Il calcolo del RUL è derivato dalla curva HI generata, rappresentando il tempo necessario per raggiungere un valore di soglia predefinito, indicativo del potenziale guasto del motore. L'approccio VAE-GAN offre vantaggi rispetto ai metodi tradizionali, presentando una maggiore precisione nella cattura delle relazioni non lineari complesse nei dati del motore. La GAN contribuisce alla generazione di curve HI realistiche e coerenti con la storia del motore, migliorando l'affidabilità delle predizioni. Inoltre, il VAE-GAN dimostra robustezza rispetto al rumore e alle anomalie nei dati, garantendo predizioni più affidabili. Il metodo rappresenta un significativo avanzamento nella manutenzione predittiva dei motori aeronautici, offrendo maggiore accuratezza, affidabilità e robustezza rispetto agli approcci tradizionali.
I modelli di diffusione rappresentano una classe di modelli innovativi che hanno seguito le GAN [34]. L'approccio principale di questi modelli è quello di generare nuovi dati a partire da un dataset casuale mediante un processo noto come "diffusione" [32]. La fase apprendimento prevede la trasformazione del dataset di training in un dataset casuale attraverso un procedimento denominato "noising", che consiste nella modifica progressiva

dei dati iniziali per un grande numero di passaggi. Dopo la fase di addestramento, i modelli di diffusione sono in grado di generare nuovi dataset applicando il processo inverso, chiamato "denoising", partendo da un dataset casuale. In pratica, ciò significa che il modello ricostruisce dati significativi da un insieme di dati casuali, dimostrando una notevole capacità di generazione. Questo tipo di algoritmi trova spesso applicazione nella generazione automatica di immagini, con Stable Diffusion che rappresenta uno dei modelli più popolari in questo contesto.

Tra gli ultimi modelli di IA sviluppati ad oggi vi sono i trasformatori [34]. Essi apportano nello sviluppo delle intelligenze artificiali concetti innovativi quali l'auto-attenzione. I trasformatori sono formati da due componenti principali: l'encoder si occupa di analizzare l'input fornito dall'utente, mentre il decoder si occupa di trasformare l'input nella risposta desiderata dall'utente. Gli encoder e decoder, all'interno di modelli di testi, condividono spesso una struttura simile. Inizialmente, essi si occupano di trasformare il testo in una rappresentazione distribuita delle parole, comunemente attraverso l'uso di tecniche di word embedding [33]. Questo processo mira a catturare le relazioni semantiche tra le parole, permettendo al modello di comprendere il significato del testo. Tuttavia, una sfida importante in questo contesto è la mancanza di informazioni sulla posizione delle parole nell'input. Per affrontare questo problema, viene introdotto il positional encoding. Questa tecnica aggiunge informazioni sulla posizione delle parole, consentendo al modello di comprendere la sequenza e la struttura temporale del testo. Per gestire le relazioni tra le parole all'interno di ogni frase, viene impiegata un'ulteriore analisi chiamata auto-attenzione. Questo processo consente al modello di attribuire pesi differenti alle parole in base alle relazioni tra loro per catturare dipendenze a lungo termine e relazioni complesse nel testo. Nel contesto dell'encoder e del decoder, l'auto-attenzione viene applicata sia alle parole dell'input (nel caso dell'encoder) che alle parole generate finora (nel caso del decoder). Ciò permette al modello di considerare le interazioni tra tutte le parole coinvolte, migliorando la qualità del testo generato. In sintesi, questa combinazione di word embedding, positional encoding e self-attention rappresenta un approccio potente per

catturare le relazioni semantiche e la struttura sequenziale nei modelli di testi. Oggigiorno, questa architettura è utilizzata nei modelli di processamento del linguaggio naturale (detti NLP), i quali sono alla base dei più noti chatbot sul mercato quali ChatGPT e Llama.
Uno studio analizza come i trasformatori emergono come una risorsa di rilievo nell'ambito della sicurezza informatica, apportando notevoli miglioramenti [11]. Questi modelli facilitano l'analisi delle relazioni all'interno di input come sequenze di testo o codice, consentendo l'identificazione delle informazioni più rilevanti. In questo contesto, le loro applicazioni nella sicurezza informatica sono molteplici. In primo luogo, possono essere impiegati per individuare malware, esaminando il codice e identificando schemi caratteristici di minacce informatiche. Ciò può avvenire attraverso il confronto con database di malware noti o analizzando il comportamento del codice in esecuzione. Inoltre, i meccanismi di attenzione si rivelano utili nel rilevare anomalie, monitorando il comportamento dei sistemi e identificando eventuali deviazioni indicative di un attacco in corso. Questo processo può coinvolgere l'analisi di log di sistema, traffico di rete e altri dati pertinenti. Un altro impiego importante consiste nella prevenzione delle intrusioni, dove tali meccanismi possono identificare vulnerabilità nei sistemi e prevenire lo sfruttamento da parte di potenziali attaccanti. Questa analisi può estendersi al codice sorgente, alla configurazione di rete e altri dati correlati. Infine, i meccanismi di attenzione del Transformer possono essere sfruttati per integrare dati provenienti da diversi sensori IoT, identificando minacce al di là delle capacità di un singolo sensore. Le tendenze attuali indicano una transizione dalla mera prevenzione degli attacchi alla predizione e risposta agli stessi. In questo contesto, i meccanismi di attenzione si rivelano strumenti preziosi, consentendo l'identificazione di segnali predittivi per adottare misure preventive tempestive. Inoltre, l'integrazione crescente di sensori IoT nella sicurezza informatica amplifica l'utilizzo di tali meccanismi. La capacità di aggregare dati da diverse fonti consente di individuare minacce al di là delle limitazioni di un singolo sensore. In conclusione, i meccanismi di attenzione dei trasformatori si rivelano una risorsa

promettente per affrontare le sfide complesse della sicurezza informatica contemporanea, grazie alle loro capacità analitiche e all'identificazione di schemi intricati.

1.2.2 Stato attuale delle IA generative

Allo stato attuale, le IA generative sono diventate molto potenti e i nuovi modelli Large Language Models sono capaci di rispondere a prompt scritti in linguaggio naturale. Attualmente il più usato è ChatGPT, il chatbot di OpenAI che fa uso della rete neurale GPT. L'utilizzo di IA generative risulta vantaggioso per molti progetti e ciò è dovuto a due loro caratteristiche importanti. In primo luogo, la loro natura di algoritmi consente di automatizzare molti processi e di ridurre i tempi. In secondo luogo, si tratta di algoritmi progettati per essere flessibili e simulare il lavoro umano grazie ai meccanismi di apprendimento, garantendo loro un'ottima capacità di adattarsi. Esempi di applicazione di IA generative in contesti industriali possono essere la progettazione di nuovi materiali, design, dataset sperimentali, parti meccaniche [35]. Attualmente, le genAI hanno maggiormente uso nell'ambito dello sviluppo software: infatti strumenti come Github CoPilot e ChatGPT sono accurati nella generazione di codice quando si tratta di richieste semplici, facendo risparmiare tempo agli sviluppatori.

Un approccio empirico [12] è stato adottato per esaminare la possibile sostituzione della programmazione a coppie umane con GitHub Copilot. Lo studio coinvolge 21 partecipanti e si è concentrato sulla realizzazione del gioco Minesweeper in Python. Durante i test, i partecipanti hanno operato in tre condizioni distintive: programmazione a coppie con l'ausilio di Copilot, programmazione a coppie umane con un partecipante svolgente il ruolo di "driver" e l'altro di "navigator" e infine, programmazione a coppie umane con ruoli invertiti. Nel contesto dell'indagine, il "driver" è il partecipante incaricato di scrivere attivamente il codice. Questa figura ha il compito di tradurre le idee e le decisioni prese dalla coppia in codice eseguibile. Il "navigator", d'altra parte, assume un ruolo più analitico, esaminando attentamente il codice scritto dal "driver". Il "navigator" si concentra sulla comprensione del codice, suggerendo possibili miglioramenti, rilevando errori e contribuendo a una visione più chiara della soluzione implementata. Le prestazioni sono

state valutate attraverso l'analisi delle righe aggiunte e rimosse durante i test. Le righe aggiunte sono state considerate un indicatore di produttività, mentre le righe rimosse sono state interpretate come un segnale di possibile degrado nella qualità del codice. I risultati hanno indicato che GitHub Copilot ha generato un numero significativamente maggiore di righe rispetto alla programmazione a coppie umane, suggerendo una maggiore produttività nell'utilizzo di Copilot. Tuttavia, sono state eliminate più righe di codice nei progetti sviluppati con Copilot, indicando che la qualità del codice generato potrebbe essere inferiore rispetto a quello prodotto dalla collaborazione umana. In conclusione, i risultati suggeriscono che GitHub Copilot potrebbe essere più produttivo in quanto genera una maggiore quantità di codice, ma al contempo potrebbe influire sulla qualità complessiva del risultato.

Un secondo studio empirico [13] è stato condotto al fine di esplorare l'efficacia di GitHub Copilot nell'automatizzare la risoluzione di problemi di programmazione nei corsi di informatica. Il focus principale è stato valutare la correttezza e la comprensibilità dei codici generati da Copilot in vari linguaggi di programmazione, inclusi Python, Java, JavaScript e C. La metodologia adottata ha previsto una selezione casuale di domande di diversa difficoltà dalla piattaforma LeetCode per un totale di 33 quesiti, seguita dalla generazione delle risposte da parte di Copilot attraverso l'ambiente di sviluppo Visual Studio Code (VS Code). I risultati sono stati successivamente sottoposti a verifiche su LeetCode mentre la comprensibilità, misurata dalla complessità ciclomatica, dei codici generati è stata calcolata utilizzando SonarQube. Dai risultati ottenuti, emerge che il linguaggio di programmazione Java è stato il più efficacemente supportato da Copilot, con una percentuale del 57% di risposte corrette. Al contrario, JavaScript ha mostrato la percentuale inferiore, con solo il 27% di risposte corrette. Inoltre, l'analisi della complessità ciclomatica media dei codici generati ha rivelato che il codice generato da Copilot è generalmente comprensibile.

Un pregio importante degli algoritmi di IA generativa è la loro conoscenza di molti argomenti anche diversi tra loro. Ciò si traduce in un’ottima versatilità in molti settori anche diversi tra loro. Nel contesto del controllo del traffico, ChatGPT è stato coinvolto in una

serie di test e valutazioni per determinare le sue capacità e potenziali utilizzi [14]. Le valutazioni si sono concentrate su diversi aspetti, tra cui l'acquisizione di conoscenze immediate per i gestori del traffico, l'analisi della topologia delle reti stradali, la previsione dei flussi di traffico e la gestione ottimale di essi, nonché la capacità di fornire suggerimenti pratici e di interpretare i codici per la gestione dei segnali stradali. Durante i test, ChatGPT ha dimostrato di possedere le conoscenze di base necessarie per comprendere i concetti fondamentali legati al controllo del traffico. È stato in grado di analizzare in modo accurato la struttura di un incrocio stradale e le sue caratteristiche, così come di individuare e gestire i flussi di traffico in maniera completa e ottimale. Inoltre, ha dimostrato di poter fornire suggerimenti utili ai gestori del traffico per migliorare l'efficienza e l'ottimizzazione del traffico. ChatGPT è stato in grado di ottimizzare il traffico e aumentare la velocità media dei veicoli nella simulazione, dimostrando così la sua capacità di contribuire in modo tangibile al miglioramento della gestione del traffico. In sintesi, ChatGPT ha dimostrato di essere una risorsa promettente nel campo del controllo del traffico, in grado di fornire analisi approfondite, suggerimenti pratici e ottimizzazione delle strategie di gestione del traffico, contribuendo così a migliorare l'efficienza e la fluidità del flusso veicolare nelle reti stradali.

Altro utilizzo derivante dalle IA generative che può avere risvolti interessanti è rappresentato da Mind Palette [15]. Mind Palette è un'innovativa applicazione di arteterapia digitale che sfrutta modelli avanzati di intelligenza artificiale generativa per offrire un supporto unico nel campo della salute mentale e dell'espressione creativa. Questa piattaforma interattiva offre un'esperienza personalizzata, guidando gli utenti attraverso diverse fasi per favorire l'esplorazione delle emozioni e la manifestazione di sé attraverso l'arte. Al suo nucleo, Mind Palette integra due potenti modelli di intelligenza artificiale: GPT-3, un sofisticato modello di linguaggio che permette conversazioni naturali, e DALL-E, un modello di generazione di immagini che consente la creazione di immagini. L'utente avvia una conversazione con il modello di IA generativa, il quale agisce come un terapeuta virtuale. Attraverso domande mirate, il sistema guida l'esplorazione dello stato emotivo

dell'utente, preparandolo per la successiva fase di disegno, durante la quale l'IA fornisce istruzioni e domande per guidare il processo. Mind Palette offre anche la possibilità di creare immagini uniche basate sulle emozioni e i pensieri dell'utente. Un elemento distintivo di Mind Palette è la generazione di consigli personalizzati da parte dell'IA, allo scopo di supportare gli utenti nel superamento di pensieri negativi e di migliorare il loro benessere mentale. L'applicazione è pensata per diversi utilizzi, rivelandosi preziosa per coloro che desiderano migliorare la propria salute mentale, per i professionisti dell'arteterapia alla ricerca di nuovi strumenti di supporto, e per gli artisti desiderosi di esplorare nuove forme di espressione creativa.

Il potenziale dell'intelligenza artificiale (IA) nel migliorare la collaborazione con gli operatori umani, tra cui quelli medici come nel caso della colonscopia, è notevole [16]. Le sfide intrinseche all'individuazione di anomalie da parte dell'essere umano, influenzate da molteplici variabili, difetti attenzionali e degrado delle prestazioni nella periferia del campo visivo, possono essere superate grazie alle capacità della IA. Differenziandosi dagli esseri umani, l'IA non è afflitta dai difetti attenzionali, fornendo così un vantaggio considerevole. I benefici derivanti dall'utilizzo dell'IA includono la capacità di evidenziare aree "sospette" durante la procedura di colonoscopia, la possibilità di identificare le cause delle anomalie e fornire feedback ai medici junior, contribuendo al loro apprendimento. Tuttavia, è cruciale riconoscere che la collaborazione efficace tra l'IA e l'essere umano richiede approcci differenti rispetto alle dinamiche umane tradizionali. Le IA più avanzate, pur offrendo prestazioni eccellenti, possono mancare di trasparenza e leggibilità, creando la necessità di strategie specifiche per migliorare le interazioni. L'utilizzo dell'IA nell'identificazione di lesioni ha dimostrato un netto miglioramento delle performance. Tuttavia, è necessario ottimizzare le interazioni tra l'IA e gli operatori umani, stabilendo protocolli chiari e addestrando i medici sulle potenzialità dell'IA, insieme a un monitoraggio continuo dei risultati. In sintesi, la promozione di una collaborazione ottimale tra l'IA e gli esseri umani richiede una comprensione approfondita delle dinamiche coinvolte e un approccio attento alla progettazione e all'implementazione di queste tecnologie avanzate nella pratica medica.

Nell'ambito delle risorse umane, l'integrazione dell'intelligenza artificiale (IA) ha rivoluzionato diversi aspetti, come la selezione dei candidati, la retention, la costruzione della carriera e la gestione complessiva delle risorse umane [17]. Un'analisi approfondita di casi di studio evidenzia l'ampio utilizzo e studio dell'IA, con particolare attenzione all'impiego di algoritmi basati su alberi decisionali e text mining nella selezione dei candidati. Attualmente, i modelli più diffusi sul mercato rientrano nella categoria dei chatbot, che si sono dimostrati strumenti efficaci per accelerare le operazioni di risorse umane. L'IA, in questo contesto, non solo ottimizza i processi, ma produce anche risultati di qualità superiore. Questo progresso è reso possibile grazie alla sinergia tra esperti di risorse umane e specialisti di IA, che collaborano per sviluppare modelli predittivi avanzati. Ad esempio, nell'ambito della retention dei dipendenti, l'IA viene utilizzata per predire l'attrito dei dipendenti, facilitando la pianificazione e la gestione delle carriere attraverso la prescrizione di attività di formazione e promozioni personalizzate. Tuttavia, l'implementazione dell'IA nei processi di risorse umane comporta anche rischi significativi. Uno dei principali è rappresentato dal bias degli algoritmi, che può portare a discriminazioni involontarie. La protezione della privacy e la sicurezza dei dati sono ulteriori preoccupazioni, poiché l'IA manipola e analizza informazioni sensibili. Affrontare questi rischi richiede una rigorosa governance, che includa meccanismi per identificare e mitigare i bias, oltre a protocolli robusti per la protezione dei dati personali. In sintesi, l'utilizzo dell'IA nelle risorse umane ha introdotto notevoli miglioramenti in termini di efficienza e precisione. Tuttavia, per massimizzare i benefici e mitigare i rischi, è essenziale un approccio integrato che coinvolga sia gli esperti di risorse umane che gli specialisti di IA.

Nell'ambito della ricerca in psicologia sociale, l'applicazione di ChatGPT offre una prospettiva intrigante sull'impiego di intelligenza artificiale generativa [18]. L'utilizzo di questa tecnologia presenta notevoli vantaggi, delineando un panorama ricco di possibilità per gli studiosi. Una delle principali utilità di ChatGPT risiede nella sua capacità di generare interazioni sociali simulando il dialogo umano. Questa caratteristica apre la strada allo

studio di scenari complessi e difficilmente realizzabili, consentendo un'analisi dettagliata delle dinamiche sociali. L'abilità dell'IA nell'interpretare e incarnare diverse personalità consente agli studiosi di esplorare una vasta gamma di comportamenti, agevolando la comprensione di meccanismi sociali complessi, e la capacità di analizzare grandi volumi di dati testuali facilita l'identificazione di trend e processi sociali macroscopici, fornendo una prospettiva ampia sulla dinamica delle interazioni umane. L'analisi testuale avanzata consente, inoltre, di identificare i processi cognitivi che sottendono al comportamento sociale, aprendo la strada a una comprensione più approfondita dei motivi alla base delle interazioni umane. La predizione dei comportamenti è un ulteriore beneficio, consentendo di anticipare sviluppi e dinamiche sociali in base alle analisi condotte. Tuttavia, è cruciale riconoscere alcuni svantaggi correlati all'impiego di ChatGPT in questo contesto, tra i quali i bias presenti nei dati di addestramento e la comprensione limitata dei contesti sociali. Inoltre, va notato che l'insorgere di allucinazioni da parte del modello e il rischio di disinformazione sono aspetti critici da considerare, soprattutto in riferimento alle implicazioni etiche della ricerca. La limitata interpretabilità dei risultati ottenuti e la scarsa creatività del modello potrebbero altresì influenzare la validità delle conclusioni raggiunte. Inoltre, occorre prestare attenzione al rispetto della proprietà intellettuale dei dati, garantendo un utilizzo etico e conforme alle normative vigenti. In conclusione, l'applicazione di ChatGPT nella ricerca sulla psicologia sociale si configura come un ambito di studio promettente, ma richiede una ponderata considerazione dei suoi limiti e delle sfide etiche connesse.

1.2.3 AI e la Business Intelligence

Le IA generative stanno avendo successo anche nel campo della Business Intelligence, in quanto superano le limitazioni dei sistemi tradizionali [36].

Attualmente, con l'avanzamento degli strumenti di analisi dati e business intelligence quali, ad esempio, Power BI di Microsoft, le aziende hanno a disposizione potenti risorse per analizzare rapidamente i dati a livelli di dettaglio desiderati attraverso la generazione di dashboards. Tuttavia, nonostante i progressi, persistono alcune sfide che limitano l'efficacia

dei processi di BI. Innanzitutto, la complessità degli strumenti attuali richiede una profonda conoscenza per sfruttarli appieno, creando una barriera all'accesso ai dati e alle informazioni che limita l'utilizzo a un pubblico ristretto. Questo aspetto rende fondamentale investire nella formazione per garantire che un numero più ampio di utenti possa beneficiare delle potenzialità di queste piattaforme. In secondo luogo, nonostante le capacità avanzate, le dashboard potrebbero non rappresentare l'intera gamma di dati disponibili, potenzialmente nascondendo informazioni cruciali per le decisioni aziendali. In aggiunta, la staticità delle dashboards può rappresentare un ostacolo significativo nei contesti aziendali dinamici, poichè l'incapacità di adattarsi rapidamente ai cambiamenti può limitare la tempestività delle decisioni e la capacità di rispondere prontamente alle mutevoli condizioni di mercato. Per affrontare questa sfida, è necessario esplorare soluzioni che consentano una maggiore flessibilità e dinamicità nelle rappresentazioni visuali dei dati.

L'implementazione di modelli di IA all'interno dei processi di BI potrebbe risolvere queste problematiche apportando, così, molti benefici.

Secondo un'analisi del 2023 [19], l'impiego dell'intelligenza artificiale (IA) nel contesto della business intelligence (BI) rivela una sinergia potente e trasformativa. La BI attraversa diverse fasi importanti: dalla raccolta e integrazione dei dati all'analisi, alla visualizzazione e alla produzione di report. In parallelo, l'IA offre un'ampia gamma di capacità che si sovrappongono con i requisiti della BI. Entrambi i settori convergono nell'obiettivo di elaborare ingenti quantità di dati per estrarre informazioni significative e guidare decisioni informate. Un elemento cruciale è l'automazione, che riduce la dipendenza da processi manuali ripetitivi e libera risorse umane per processi di maggior valore. L'analisi avanzata è un'area in cui l'IA si distingue particolarmente, consentendo la scoperta di pattern, correlazioni e trend nei dati che possono rimanere altrimenti nascosti. La BI tradizionale spesso si limita all'analisi descrittiva, mentre l'IA permette di progredire verso previsioni e suggerimenti prescrittivi, ampliando così la portata delle decisioni supportate dai dati. L'IA può migliorare ulteriormente il processo di BI offrendo analisi in tempo reale, consentendo

agli utenti di ottenere informazioni aggiornate e reattive alle situazioni di business in evoluzione. L'interfaccia utente rappresenta un'altra area di miglioramento, con la possibilità di impiegare chatbot capaci di processare il linguaggio naturale per rendere gli strumenti di BI più accessibili anche a coloro che non sono esperti nel settore. L'IA può identificare in modo autonomo gli strumenti di visualizzazione dei dati più adatti al contesto specifico, migliorando l'efficacia della comunicazione visiva dei risultati. Inoltre, può individuare anomalie e outliers nei dati, fornendo un ulteriore livello di insight e supportando la rilevazione tempestiva di situazioni anomale o problematiche. In sintesi, l'integrazione dell'IA nella BI offre una serie di vantaggi che vanno dalla generazione di insights più approfonditi e predittivi alla semplificazione dell'interazione utente e all'ottimizzazione dei processi decisionali aziendali.

Secondo lo studio condotto da Tcukanova et al. [20], l’'integrazione dell'intelligenza artificiale (IA) rappresenta un fondamentale progresso nell'evoluzione dei sistemi di Business Intelligence (BI), introducendo nuove prospettive per l'analisi e l'utilizzo dei dati aziendali. Le capacità intrinseche di apprendimento automatico e analisi avanzata dell'IA contribuiscono in modo significativo a una comprensione più approfondita del contesto aziendale, facilitando decisioni più informate e ottimizzando i processi operativi. In termini di modalità di integrazione dell'IA nei sistemi BI, è possibile adottare due approcci principali. In primo luogo, l'inserimento diretto di algoritmi IA nel codice dei sistemi BI emerge come una soluzione che non solo accelera l'esecuzione di varie attività software ma fornisce anche supporto agli utenti meno esperti, oltre a automatizzare processi di machine learning. In alternativa, la creazione di script Python indipendenti offre una maggiore flessibilità, superando le limitazioni delle funzionalità predefinite del sistema BI e consentendo l'esplorazione di un ampio spettro di possibilità. Le ricadute positive dell'integrazione dell'IA sono rilevanti. La capacità di supporto del linguaggio naturale consente di interrogare i dati in modo intuitivo, mentre assistenti digitali e chatbot forniscono interfacce interattive per un accesso agevole alle informazioni. Inoltre, l'implementazione di sistemi di suggerimenti agevola gli utenti nell'individuare dati

rilevanti per le loro specifiche esigenze. Tuttavia, è fondamentale considerare anche gli svantaggi associati a questa integrazione, in quanto l'utilizzo dell'IA richiede hardware più performante, e la verifica accurata dei risultati generati diventa cruciale. In conclusione, mentre l'integrazione dell'IA nei sistemi BI offre notevoli vantaggi, la sua adozione richiede una valutazione accurata delle opzioni disponibili e la considerazione attenta di potenziali ostacoli. L'implementazione di questa tecnologia può catalizzare un avanzamento significativo nell'analisi dei dati aziendali, contribuendo a un processo decisionale più efficiente e a una maggiore competitività sul mercato.

1.2.4 Rischi e problematiche legate alle IA generative

Oltre ai benefici apportati dalle Intelligenze Artificiali generative, è necessario tenere conto delle potenziali problematiche che potrebbero emergere [37].

Uno studio focalizzato sulla rivelazione delle sfaccettature negative dell'impiego di decisioni basate sull'intelligenza artificiale (IA) in contesti business-to-business (B2B) è stato condotto attraverso interviste individuali [21]. Gli intervistati, impiegati di un'azienda nel settore energetico specializzata nella vendita di energia, hanno delineato diverse problematiche legate all'implementazione dell'IA. Una delle principali preoccupazioni riguarda l'interazione con i clienti, dove l'impiego di chatbot IA non riesce ad adattarsi efficacemente alle esigenze individuali, portando a una diminuzione della soddisfazione dei clienti. Inoltre, sono state evidenziate serie preoccupazioni riguardanti la sicurezza e la privacy dei dati, con il rischio di violazioni o compromissioni dei dati sensibili aziendali. Tra le problematiche emerse, vi è stata anche una tendenza verso aspettative errate sulle prestazioni dei modelli IA, insieme a un potenziale "deskilling" delle risorse umane, le quali potrebbero diventare troppo dipendenti dall'IA, con conseguenti tensioni dovute al rischio di disoccupazione o alla riorganizzazione delle risorse umane all'interno dell'azienda. Ulteriore questione è la difficoltà nell'individuare i responsabili delle decisioni da parte del modello di IA tra gli sviluppatori e gli utenti finali, nonché conflitti di interesse sia tra gli sviluppatori dei modelli IA e gli impiegati, sia tra coloro che utilizzano l'IA e coloro che non lo fanno, insieme a potenziali conflitti gerarchici all'interno dell'azienda.

Un altro problema deriva dalla natura unsupervised dei modelli, per cui le soluzioni proposte non sono riproducibili o interpretabili. Per affrontare le sfide sono stati sviluppati diversi meccanismi, che includono l'implementazione di reti neurali sparse e l'uso di mappe di calore. Le reti neurali sparse sono reti in cui molte connessioni tra neuroni sono impostate a zero durante l'addestramento. Di conseguenza, si ottiene una rete con meno parametri e una struttura più comprensibile. Inoltre, le reti neurali sparse possono anche essere più efficienti in termini di calcolo, poiché richiedono meno risorse computazionali per l'addestramento e l'esecuzione. Le mappe di calore sono utilizzate per visualizzare e interpretare l'importanza relativa delle diverse caratteristiche di input per il modello neurale. Ad esempio, in un contesto di classificazione di immagini, una mappa di calore potrebbe evidenziare quali parti dell'immagine sono più influenti nel determinare la classe predetta dal modello. Questo aiuta gli utenti a comprendere quali informazioni l'algoritmo sta utilizzando per prendere decisioni.

Nella ricerca attuale, è stato proposto un approccio innovativo basato su Generative Adversarial Networks (GAN) per la rilevazione delle malattie del torace attraverso radiografie, fornendo simultaneamente una spiegazione delle decisioni del modello [22]. L'obiettivo principale è quello di generare maschere binarie esplicative delle aree affette da patologie polmonari senza richiedere un addestramento specifico per questa attività. Il modello proposto si compone di due generatori e due discriminatori. Il primo generatore è responsabile della creazione della "disease map" a partire dalle radiografie di pazienti malati. La "disease map" rappresenta graficamente le regioni dei polmoni che sono affette dalla malattia, e viene convertita in una "binary mask" per evidenziare in modo chiaro e definito le zone interessate dalla patologia. Questo processo consente di identificare e isolare le aree malate nelle immagini radiografiche. Successivamente, mediante la sottrazione della "disease map" dalla radiografia originale, si ottiene una seconda radiografia che rappresenta un paziente sano. Questo passaggio è cruciale per la creazione di un set di dati equilibrato per l'addestramento del discriminatore. Il primo discriminatore, invece, si occupa di distinguere le radiografie di pazienti sani reali da quelle generate

durante il processo di sottrazione, garantendo così che le immagini simulate siano realistiche e coerenti con le radiografie reali. Il secondo generatore sfrutta le radiografie dei pazienti sani generate per creare la controparte delle radiografie di pazienti malati. Infine, il secondo discriminatore opera analogamente al primo, ma si concentra sul distinguere le radiografie di pazienti malati reali da quelle generate dal secondo generatore. In sintesi, il modello proposto utilizza una combinazione di GAN per generare maschere binarie esplicative delle malattie polmonari, rendendo il processo di rilevamento delle patologie più interpretabile e automatizzato.

1.2.5 Ultime tendenze e sfide future delle IA generative

Il settore delle IA generative è relativamente recente ed è esploso solo nel 2023. Nel tempo si prevedono nuove tendenze dovute alla recente spinta [38].

Nel corso del tempo, ci si aspetta che emergano nuovi modelli di intelligenza artificiale sempre più avanzati e precisi. Oltre agli attuali Large Language Model (LLM), si prospetta la proliferazione di algoritmi di intelligenza artificiale generativa ad hoc, specificamente addestrati per contesti particolari. Questa evoluzione potrebbe portare a soluzioni più efficienti e specializzate per rispondere alle esigenze specifiche di settori o applicazioni particolari.

Attualmente, i tool di intelligenza artificiale generativa sono in grado di produrre contenuti audio e visivi, sebbene non raggiungano ancora il livello di sofisticazione degli algoritmi di generazione di testi. Tuttavia, con la crescente diffusione online di contenuti multimediali in forma di video e audio, è plausibile che i prossimi strumenti di intelligenza artificiale generativa dedicati a queste forme di media saranno sviluppati e adottati in modo simile alle controparti focalizzate sulla generazione di testi. Inoltre, mentre attualmente molti tool sono specializzati in un formato di risposta specifico, c'è la possibilità che emergano strumenti multimodali in grado di generare non solo testi, ma anche video e audio in risposta alle richieste degli utenti. Questa evoluzione potrebbe portare a un'interazione più ricca e diversificata con le tecnologie di intelligenza artificiale, consentendo una maggiore

flessibilità nell'espressione e nella comprensione delle informazioni attraverso diversi canali sensoriali.

Il campo degli agenti autonomi rappresenta un ambito di sviluppo promettente per le intelligenze artificiali, con la prospettiva di ottenere risultati significativi. Gli agenti autonomi sono algoritmi di intelligenza artificiale generativa in grado di gestire richieste complesse scomponendole in sotto-processi più gestibili. Attualmente, uno dei più noti in questo ambito è AutoGPT, che fa uso di GPT-4. Benché soggetto a numerosi errori in questo stadio, si prevede che nel tempo gli agenti autonomi saranno sviluppati in maniera sempre più efficace e performante, consentendo loro di affrontare processi sempre più complessi con maggiore precisione e affidabilità.

Insieme allo sviluppo e al miglioramento degli algoritmi di IA generativa, ci si chiede quali possano essere i futuri risvolti sociali di questa tecnologia.

Nel contesto attuale caratterizzato dalla presenza di modelli generativi e di vasti algoritmi, è imperativo affrontare le sfide legate ai deepfake, alla disinformazione [39] e alla manipolazione dell'opinione pubblica. Stando a quanto riportato in un’analisi [23], i pericoli della disinformazione includono impatti indesiderati sull'opinione pubblica, erosione della democrazia, minacce alla privacy e sicurezza personale, riduzione della fiducia nei media e nascita di nuovi dilemmi legali. Questi fenomeni sono amplificati dalla rapida evoluzione degli algoritmi di IA generativa, rendendo necessaria una collaborazione tra istituti di ricerca, aziende private e governi per sviluppare soluzioni efficaci. Le possibili soluzioni tecniche comprendono algoritmi di riconoscimento dei contenuti falsi e metodologie di autenticazione per distinguere i contenuti generati da IA. Tuttavia, l'evoluzione rapida degli algoritmi generativi presenta sfide nell'implementazione di tali soluzioni. Pertanto, è cruciale adottare un approccio integrato, coinvolgendo social media, istituti educativi e organizzazioni governative. Le politiche per mitigare la disinformazione includono il coinvolgimento attivo dei social media nella moderazione dei contenuti, la collaborazione con fact-checkers indipendenti, meccanismi di segnalazione da parte degli utenti, trasparenza sui risultati e formazione degli utenti. Inoltre, la formazione delle

persone attraverso programmi di alfabetismo digitale, sviluppo del pensiero critico e promozione di contenuti verificati è essenziale. L'implicazione etica della disinformazione generata dalle IA richiede che i creatori di modelli minimizzino i bias, garantiscano trasparenza nello sviluppo e nel funzionamento dei modelli e si rendano responsabili nei confronti del pubblico. La gestione dei compromessi tra l'innovazione dei modelli e la mitigazione dei rischi è cruciale, insieme alla necessità di sviluppare i modelli considerando le responsabilità etiche. In conclusione, un framework integrato dovrebbe essere basato su soluzioni tecnologiche, iniziative strategiche delle aziende, sviluppo di linee guida, politiche e campagne di sensibilizzazione. Meccanismi di monitoraggio, controllo, valutazione e feedback sono essenziali, mentre è cruciale gestire i rischi etici e trovare un equilibrio tra la libertà di espressione e la protezione dalla disinformazione. La definizione di protocolli per rispondere prontamente alle eventuali conseguenze negative è un elemento chiave del processo.

Un articolo dell'UNESCO [44] sottolinea la necessità di regolare l'uso delle intelligenze artificiali generative nelle scuole, almeno in fase iniziale. Questa prospettiva si basa sull'idea che l'educazione pubblica dovrebbe rimanere un atto "umano", e che le IA generative potrebbero non essere in grado di sostituire adeguatamente l'interazione umana nell'insegnamento. L'articolo fa riferimento alla dimostrazione di questa limitazione durante i lockdown causati dalla pandemia COVID-19, quando la transizione alla didattica a distanza ha portato a un peggioramento dei risultati accademici degli studenti. La regolamentazione proposta ha lo scopo di garantire che l'implementazione delle intelligenze artificiali generative nelle scuole avvenga in modo responsabile, considerando le sfide e i limiti attuali. Questo approccio deriva dalla consapevolezza che, almeno per ora, le competenze umane, come la comprensione emotiva e la capacità di adattarsi a situazioni complesse, sono fondamentali nell'ambito educativo e che attualmente non sono completamente sostituibili da algoritmi.

La legislazione e la regolamentazione dell'utilizzo degli strumenti di intelligenza artificiale rappresentano una sfida significativa per la società [40]. La rapida espansione di questo

settore complica la definizione di normative adeguate e aggiornate in modo tempestivo. Le difficoltà legate a questa dinamica evolutiva includono la necessità di tenere il passo con i miglioramenti e le novità degli algoritmi di intelligenza artificiale generativa. Inoltre, emergono questioni etiche cruciali, come la trasparenza nell'uso delle informazioni derivanti dall'utilizzo di intelligenza artificiale generativa. Un esempio è la necessità di stabilire se le aziende debbano informare i clienti sull'impiego di tali tecnologie e sulle conseguenze legate all'elaborazione dei loro dati. Questi aspetti etici diventano sempre più rilevanti, poiché la diffusione dell'IA influisce sempre più su vari aspetti della vita quotidiana, dall'assistenza sanitaria alla sicurezza, e richiede un equilibrio tra l'innovazione tecnologica e la necessità che gli individui possano fare una scelta informata. La definizione di normative adeguate diventa quindi cruciale per garantire un utilizzo responsabile ed etico dell'intelligenza artificiale generativa.

La diffusione massiccia delle intelligenze artificiali generative nel mainstream presenta un rischio significativo dovuto all’effetto gregge. La velocità con cui gli algoritmi operano, combinata alla competitività dei mercati, potrebbe spingere le aziende a implementare rapidamente l'IA nei loro processi prima che siano consolidati adeguati meccanismi di controllo dei risultati. Questo scenario potrebbe esporre le aziende e i professionisti al pericolo di generare contenuti errati, distorti o di bassa qualità, con conseguenti impatti negativi sui ricavi e sulla reputazione. La pressione per adottare rapidamente le tecnologie può portare a una mancanza di attenzione alla qualità e all'accuratezza dei risultati prodotti dalle intelligenze artificiali. Ciò potrebbe avere conseguenze gravi, sia in termini di perdita di fiducia da parte dei clienti che nell'ambito delle performance aziendali. Pertanto, è fondamentale bilanciare la velocità di adozione dell'IA con la necessità di implementare rigorosi controlli di qualità e garantire che i professionisti e le aziende comprendano appieno i limiti e le responsabilità legate all'utilizzo di tali tecnologie.

1.3 Generative ETL

1.3.1 Problematiche attuali dei processi ETL

L'ETL è un processo intrinsecamente complesso e costoso da realizzare. Questo è principalmente dovuto alla diversità delle esigenze progettuali dei clienti, che rende difficile creare una soluzione unica per tutti e quindi automatizzare completamente il processo. Inoltre, i cambiamenti interni all'azienda, come ad esempio modifiche alla struttura dei dati nei database operativi, richiedono aggiustamenti continui, aumentando la necessità di monitoraggio costante dei processi ETL e quindi i costi associati.

Attualmente, la progettazione e la manutenzione dei processi ETL sono spesso gestite manualmente, con la necessità di formare il personale del cliente per gestire il processo in seguito. Questo approccio comporta diversi svantaggi, tra cui costi elevati legati alla gestione delle infrastrutture e del personale, rischi di errori umani dovuti alla ripetitività delle attività e problemi di qualità dei dati elaborati.

Inoltre, la gestione dei dati non strutturati rappresenta una sfida aggiuntiva [24]. Mentre i dati strutturati possono essere gestiti più facilmente in un flusso ETL, i dati non strutturati richiedono un lavoro supplementare e spesso vengono trascurati. Questo può portare alla perdita di informazioni preziose e compromettere la qualità delle decisioni aziendali basate sull'analisi dei dati.

Un'altra complicazione è rappresentata dalla velocità con cui vengono generati nuovi dati e dalle data stream. È essenziale catturare i dati in tempo reale per garantire che siano sempre aggiornati. Tuttavia, questo richiede un coordinamento accurato tra le operazioni di estrazione, trasformazione e caricamento dei dati, aggiungendo ulteriori complessità al processo ETL tradizionale [3,2].

1.3.2 Vantaggi delle IA generative nei contesti ETL

L'impiego delle intelligenze artificiali generative potrebbe rivestire un ruolo significativo nella progettazione e gestione delle pipeline ETL, presentando notevoli vantaggi su due fronti principali. Innanzitutto, la loro altissima flessibilità consente un adattamento rapido

alle mutevoli esigenze e la capacità di trattare dati non strutturati in maniera efficiente. Questa caratteristica risulta fondamentale in un contesto dinamico in cui le condizioni e i requisiti dei dati possono evolvere con celerità.

In secondo luogo, l'automatizzazione fornita dalle generative AI riduce la dipendenza dal lavoro umano, contribuendo a minimizzare i tempi di esecuzione e mitigare il rischio di errori umani. L'automazione di processi ripetitivi e laboriosi consente di aumentare l'efficienza operativa complessiva, liberando risorse umane per attività più ad alto valore aggiunto.

Gli impatti positivi derivanti dall'adozione di questo approccio in ambito aziendale sarebbero diversi e significativi. In primo luogo, la riduzione dei costi associati alla manodopera rappresenterebbe un beneficio tangibile, consentendo alle aziende di ottimizzare le risorse finanziarie. Inoltre, la minimizzazione degli errori umani porterebbe a una diminuzione dei costi correlati a correzioni e rettifiche, contribuendo a migliorare la qualità complessiva dei processi aziendali.

Uno dei possibili utilizzi delle IA generative nei processi ETL è la gestione automatica dei dati non strutturati. Questa capacità consente di affrontare la crescente mole di informazioni non organizzate, consentendo una maggiore efficacia nell'analisi e nell'utilizzo di tali dati per fini decisionali.

L'automazione delle attività legate alle pipeline ETL tramite generative AI può migliorare la precisione e l'efficacia delle operazioni di gestione dei dati, riducendo il rischio di scelte basate su informazioni incomplete o errate.

Un articolo recente ha approfondito le molteplici applicazioni di tali modelli, evidenziando le possibilità offerte nelle fasi di estrazione, trasformazione e caricamento dei dati [43]. Tuttavia, l’autore afferma che l'implementazione di tali soluzioni non è priva di sfide e rischi. Tra i fattori tecnici da considerare, la tutela della privacy dei dati gioca un ruolo cruciale. L'accesso e la manipolazione automatica dei dati devono essere attentamente regolamentati per garantire il rispetto delle normative sulla privacy e la sicurezza delle informazioni sensibili. Inoltre, il monitoraggio in tempo reale delle performance è

fondamentale per garantire l'efficacia e l'efficienza delle soluzioni basate su IA generativa nei processi ETL. Questo monitoraggio consente di identificare tempestivamente eventuali anomalie o problemi nelle operazioni, consentendo interventi correttivi tempestivi e ottimizzando le prestazioni complessive del sistema.

1.3.3 Stato attuale delle generative ETL

Attualmente, i casi di implementazione dei modelli di IA generativa nei processi ETL sono pochi e non ancora sviluppati appieno, ma promettenti.

Un esempio di questo è NewFangled, la quale è una piattaforma innovativa che promette un grande sviluppo nel settore tecnologico [41]. Si tratta di una soluzione no-code progettata per essere accessibile agli utenti di tutti i livelli di competenza, offrendo un assistente IA avanzato che è in grado di tradurre richieste formulate in linguaggio naturale in flussi di lavoro ETL. Questo significa che gli utenti possono comunicare con l'assistente utilizzando linguaggio comune anziché codice tecnico complesso, semplificando notevolmente il processo di sviluppo e gestione dei flussi di lavoro di estrazione, trasformazione e caricamento dei dati. L'elemento distintivo di NewFangled è la capacità del suo motore IA di apprendere in modo continuo dalle interazioni con gli utenti. Questo significa che l'assistente IA diventa sempre più efficace nel comprendere le esigenze degli utenti nel tempo, adattando e ottimizzando i flussi di lavoro per soddisfare al meglio le loro esigenze specifiche. Questa caratteristica contribuisce a rendere l'esperienza dell'utente più fluida e personalizzata, migliorando la produttività e riducendo la necessità di interventi manuali.

In un'intervista con il COO di Estera, Jay Mishra, si è approfondito il modo in cui l'azienda utilizza algoritmi di intelligenza artificiale generativa per ottimizzare la progettazione delle pipeline ETL [42]. Si è evidenziato che tali algoritmi sono particolarmente efficaci nel trattamento di dati non strutturati, sebbene per i dati strutturati l'utilità dell'IA generativa sia limitata. Tra i processi affidati a questi algoritmi vi è la generazione di codici per le trasformazioni dei dati, la creazione di mapping, compresi quelli semantici, e la definizione dello schema a stella. Un punto saliente dell'intervista è stato l'enfasi posta sull'automazione

dei processi ripetitivi e sulla correzione delle pipeline esistenti mediante l'utilizzo dei modelli di IA. Questo non solo libera i data engineers da processi monotoni, consentendo loro di concentrarsi su attività di maggiore valore, ma permette anche di ottenere risultati più efficaci nella gestione e creazione dei metadati. Estera attualmente utilizza sei modelli di IA differenti, tra cui Llama e GPT, con la possibilità di affinare le prestazioni attraverso il fine-tuning. Tuttavia, è stato sottolineato che l'IA è da considerarsi come un assistente che richiede una supervisione umana costante per garantire la correttezza dei risultati. L'implementazione dell'IA è stata finora circoscritta alla fase di progettazione e implementazione delle pipeline ETL, escludendo le esecuzioni. Tuttavia, si prevede un'estensione del suo impiego nel monitoraggio dei flussi di lavoro esistenti, con l'obiettivo di arrivare a una gestione in tempo reale dei processi ETL. Tra le sfide che Estera deve affrontare per implementare l'IA vi sono la definizione di una strategia coerente, l'identificazione delle aree in cui l'IA può apportare il maggior valore, il training del personale e la valutazione dei dati adatti al trattamento mediante IA. Inoltre, si sono evidenziate problematiche legate alla comprensibilità dei risultati e alla precisione dei modelli, soprattutto nei processi non deterministici.

2 Descrizione del progetto

Il tool sviluppato nell'ambito della tesi rappresenta un passo significativo verso l'automatizzazione del processo di progettazione e implementazione di pipeline ETL. La sua principale funzione è quella di semplificare e accelerare il processo di creazione di una pipeline ETL. La sua concezione è guidata da un approccio integrato che sfrutta sia l'analisi della struttura del database sorgente che il contributo di un algoritmo di intelligenza artificiale generativa.

Nel secondo capitolo, si procederà inizialmente con una descrizione qualitativa dell'algoritmo, illustrando il suo principio di funzionamento. Nella sezione successiva, verrà condotta una valutazione preliminare di fattibilità contenente analisi dei requisiti, analisi degli stakeholders, analisi SWOT e analisi dei benefici.

2.1 Descrizione generale dell'algoritmo

Il tool si occupa di recuperare tutte le informazioni necessarie per la progettazione del processo ETL a partire dalla struttura del database sorgente e da una sua descrizione generale, dei suoi scopi e di eventuali richieste aggiuntive da parte dell'utente sotto forma di prompt. Forniti i dati all'algoritmo di IA generativa, esso si occupa di individuare i parametri fondamentali per la progettazione del processo ETL.

Innanzitutto, il tool esplora la struttura del database sorgente per identificare tabelle, colonne e relazioni, acquisendo una comprensione dettagliata della natura dei dati. Successivamente, richiede una descrizione generale del database all'utente, incorporando scopi specifici e requisiti speciali per personalizzare la pipeline ETL in base al contesto.

Una volta ottenute tutte le informazioni necessarie, il tool fornisce i dati a un algoritmo di intelligenza artificiale generativa. Questo algoritmo analizza le informazioni e genera automaticamente i parametri essenziali per la progettazione della pipeline ETL.

Infine, basandosi sui parametri generati, il tool crea la pipeline ETL in modo automatico, eseguendo attività di estrazione, trasformazione e caricamento dei dati in conformità con le specifiche raccolte durante le interazioni precedenti.

2.2 Analisi funzionale e potenziali benefici

L'approccio globale del tool, che integra l'analisi strutturale del database con l'intelligenza artificiale generativa, mira a semplificare significativamente il processo di progettazione delle pipeline ETL. Inoltre, l'approccio automatizzato permette di ridurre i tempi di sviluppo, e l'utilizzo di algoritmi di intelligenza artificiale generativa permette di adattare la pipeline alle esigenze specifiche del contesto. Per effettuare le analisi, è stato preso come contesto di riferimento un'azienda di consulenza informatica interessata a utilizzare il progetto per velocizzare il processo di progettazione e gestione dei processi ETL richiesti dai clienti.

2.2.1 Analisi degli stakeholders

Stakeholders interni:

- dirigenti e managers, che hanno interesse nella qualità dei dati in modo da intraprendere decisioni correttamente informate;
- amministratori dei sistemi ETL, che hanno la responsabilità sul corretto funzionamento dei sistemi ETL;

Stakeholders esterni:

- CTO dell'azienda cliente, che vogliono verificare che la tecnologia sia in linea con le scelte tecnologiche e strategiche intraprese dall'azienda.
- data scientists e data analysts dell'azienda cliente, i quali hanno bisogno che il progetto funzioni correttamente per garantire la qualità dei dati sui quali lavorano;
- amministratori dei database dell'azienda cliente, i quali hanno interesse nel mantenere integre le loro architetture;
- sviluppatori di sistemi di intelligenza artificiale generativa, i quali possono fornire eventuali funzionalità aggiuntive per facilitare le operazioni sui dati;

- fornitori di software ETL, che hanno interesse a semplificare le operazioni per l'utente attraverso interfacce supportate da IA generativa;
- fornitori di servizi cloud, che vogliono mantenere alte le prestazioni delle loro architetture qualora si dovessero svolgere le operazioni di elaborazione e trasferimento dei dati con i loro servizi;
- enti pubblici, interessati agli aspetti legali del software quali garanzia della privacy e sicurezza.

2.2.2 Analisi dei requisiti

Requisiti funzionali:

- connessione al DB sorgente, ciò serve per effettuare le operazioni di trasferimento dei dati ed eseguire le operazioni di esplorazione preliminare dei dati;
- individuazione automatica dei parametri necessari per la progettazione del processo ETL e conseguente possibilità di conferma o correzione da parte dell'utente;
- creazione ed esecuzione automatica delle query SQL per l'esecuzione del processo ETL;
- generazione automatica delle tabelle qualora fosse disponibile solo la tabella sorgente, suggerimento dei nomi delle nuove tabelle a partire dal nome della tabella sorgente;
- possibilità di personalizzazione della pipeline prima e dopo la generazione, l'utente deve avere la possibilità di modificare il processo a proprio piacimento qualora dovessero avvenire dei cambiamenti.

Requisiti non funzionali:

- velocità delle operazioni sui dati, requisito fondamentale che può portare un vantaggio competitivo significativo a chi implementa il progetto;
- velocità di risposta da parte del sistema di intelligenza artificiale, la generazione veloce delle risposte favorisce la velocità dell'intero processo;

- usabilità del programma, gli utenti meno esperti devono avere la possibilità di generare il processo ETL senza dover conoscere gli aspetti più tecnici legati ad essi e in linguaggio naturale;
- versatilità e possibilità di operare su più piattaforme, è necessario che l'algoritmo funzioni su piattaforme diverse in modo da adattarsi a contesti diversi;
- accuratezza dei risultati forniti dall'IA, è necessario che la risposta fornita dal modello sia in linea con le richieste dell'utente e che non vengano generati errori tecnici.

2.2.3 Analisi SWOT

Punti di forza:

- velocità di generazione delle pipeline ETL: il processo di generazione è velocizzato e l'utente del data warehouse ha a disposizione il processo ETL anticipatamente rispetto ai metodi tradizionali;
- riduzione dei tempi e dei costi: la velocizzazione dei lavori consente di ridurre le ore di lavoro umano, ciò si traduce in una riduzione sostanziale dei costi e dei tempi legati al lavoro;
- flessibilità nella progettazione: rispetto agli algoritmi statici, l'utilizzo dell'IA generativa nella generazione della pipeline consente di adattare in maniera automatica la progettazione alle richieste specifiche dell'utente, al contesto di utilizzo e alla struttura dei dati;
- facilità di accesso: gli utenti meno esperti hanno la possibilità di automatizzare la progettazione della pipeline utilizzando il linguaggio naturale;
- possibilità di trattamento dei dati non strutturati: la capacità dei modelli di IA di analizzare dati non strutturati quali testi, immagini, video e audio consente l'accesso e l'analisi di una grande quantità di dati al momento inaccessibile.

Punti di debolezza:

- comprensibilità dei risultati: la natura black box dei modelli di IA generativa più sofisticati può generare preoccupazioni riguardo alle modalità di generazione della risposta;
- rischio di errori: è necessario effettuare un controllo preliminare sui risultati generati dal modello IA prima di accettarli, in quanto i modelli IA possono generare errori o allucinazioni.

Opportunità:

- miglioramento delle capacità degli algoritmi di IA generativa: il miglioramento continuo dei dataset di apprendimento e degli algoritmi di IA generativa permettono la generazione di risposte sempre più soddisfacenti, favorendo la correttezza dei risultati del progetto;
- aumento notevole della popolarità degli algoritmi di IA generativa: l'attuale popolarità di modelli di IA generativa quali ChatGPT e Github Copilot potrebbe portare a miglioramento dell'immagine per le aziende che ne sanno fare uso;
- costi e tempi di gestione e progettazione dei processi ETL attualmente alti: la generazione automatica di pipeline ETL e l'ulteriore riduzione dei tempi di studio del contesto aziendale e del dataset a disposizione può velocizzare i tempi di consegna congiuntamente a una riduzione del prezzo dovuta alla diminuzione dei costi, generando un vantaggio competitivo sul mercato.

Minacce:

- mercato delle tecnologie digitali altamente innovativo: il progetto opera in un mercato altamente innovativo e competitivo, dove le aziende sono in grado di raggiungere la competizione e annullare in breve tempo ogni vantaggio competitivo legato all'innovazione tecnologica. I colossi del settore quali Google possono sviluppare soluzioni allo stesso scopo di qualità più alta;
- rischio di normative nazionali o internazionali atte a limitare l'utilizzo di modelli di intelligenza artificiale.

2.2.4 Analisi dei benefici

Il progetto di sviluppo di una nuova pipeline ETL si articola attraverso diverse fasi. Inizialmente, esso si avvia con un'attenta analisi del contesto aziendale, comprendendo le esigenze specifiche e i requisiti del sistema. Questa fase consente di identificare la soluzione ottimale per l'azienda e stabilire le metodologie più adeguate da impiegare.

Successivamente, si procede con l'analisi approfondita della struttura dei dati da integrare. Questo passo è essenziale per individuare gli aspetti più tecnici della progettazione, come la definizione della struttura delle tabelle e la comprensione dettagliata delle fonti dati coinvolte.

Dopo le analisi preliminari, viene avviata la fase di disegno architetturale, nella quale si ha la progettazione teorica dei dati e della loro struttura. Vengono, inoltre, progettati i mappings e le tabelle.

Si avvia così la fase di sviluppo effettiva, durante la quale vengono generate le tabelle necessarie e si definiscono le mappature tra i dati. Tale processo è cruciale per garantire un flusso efficiente e coerente dei dati attraverso la pipeline.

Si giunge poi alla fase di testing. Durante l'esecuzione, la pipeline ETL viene implementata secondo le specifiche progettuali. Il controllo è un passaggio fondamentale per verificare la corretta

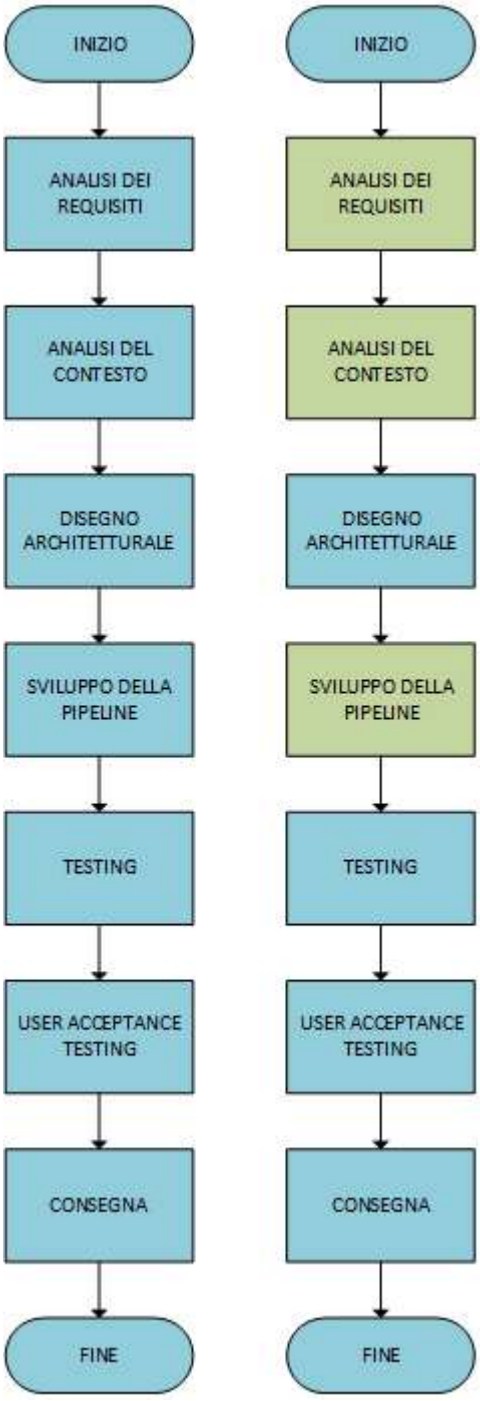

FIGURA 5. PROCESSO DI SVILUPPO DI UN PROCESSO ETL PER UN CLIENTE. LE ATTIVITÀ AUTOMATIZZATE DAL MODELLO DI IA SONO EVIDENZIATE IN VERDE

esecuzione della pipeline e per garantire che i dati siano trasformati e caricati correttamente.

Infine, viene mostrato il progetto finito al cliente per un controllo finale e per effettuare eventuali modifiche finali desiderate da esso e, se il cliente è soddisfatto, si ha la fase finale di consegna del progetto.

Gli algoritmi statici, se dotati dei parametri essenziali, possono certamente automatizzare la fase di implementazione pratica del processo ETL. Tuttavia, la loro limitazione risiede nel fatto che non sono in grado di estrarre in modo autonomo le variabili di progettazione direttamente dall'analisi del contesto aziendale.

L'inserimento di un modello di IA generativa nella prima fase del processo ETL ha il potenziale di automatizzare l'identificazione delle variabili di progettazione attraverso l'analisi del contesto aziendale. In tal modo, le parti più laboriose del processo ETL diventano suscettibili di automazione, permettendo alla IA generativa di svolgere un ruolo chiave nella progettazione.

3 Sviluppo del progetto

Nella seguente sezione si espone, in maniera più dettagliata e tecnica, il prototipo di generative ETL realizzato dallo studente. Ciò comprende l'insieme di funzionalità e scelte progettuali che lo caratterizzano.

Nel terzo capitolo, il tesista descriverà gli aspetti tecnici del primo strumento da lui sviluppato, che esegue il processo di ETL utilizzando esclusivamente i linguaggi Python e SQL, illustrando il funzionamento dei singoli componenti ed esponendo le scelte progettuali adottate. Successivamente, sarà presentato il secondo progetto realizzato in gruppo durante il tirocinio curriculare presso Mediamente Consulting srl. Quest'ultimo prevede la creazione di un file XML finalizzato alla generazione di un progetto per SQL Server Integration Services (SSIS), un software proprietario di Microsoft impiegato per la progettazione dei processi di integrazione dati. Infine, verranno confrontate le due soluzioni, evidenziando i vantaggi e le debolezze di ognuno.

3.1 Generazione operazioni ETL tramite Python

3.1.1 Scelte progettuali

Il prototipo consiste di un insieme di script che sfruttano diversi moduli per poter lavorare sui dati, realizzare i prompt da mandare al modello di IA ed eseguire su database le query SQL necessarie. Gli script sono stati realizzati con il linguaggio di programmazione Python non solo per la sua accessibilità, ma anche perché possiede le librerie necessarie a fare operazioni di analisi dei dati quali pandas e numpy.

Il modello di intelligenza artificiale generativa utilizzato è il GPT-3.5, implementato attraverso l'API della società OpenAI. Questa scelta è stata motivata principalmente dalla familiarità del tesista con il modello e dalle sue prestazioni superiori rispetto alla media. Rispetto ad altri modelli open source, il GPT-3.5 eccelle nel comprendere il linguaggio naturale e nel riconoscere più facilmente la lingua utilizzata nei prompt ricevuti. Queste caratteristiche lo rendono particolarmente adatto al progetto in questione.

L'utilizzo del modello GPT-3.5 non esclude affatto la possibilità di integrare modelli open source, specialmente considerando che il divario di prestazioni tra modelli open source e modelli di lingua come GPT-3.5 è gestibile e potrebbe addirittura ridursi in futuro con il continuo sviluppo e miglioramento della tecnologia.

Per quanto riguarda le operazioni sui dati, sono state impiegate le librerie Pandas e NumPy per Python in quanto sono una scelta solida e diffusa. Queste librerie consentono di elaborare grandi volumi di dati in forma di dataframes e array, semplificando così la realizzazione di tutte le operazioni tipiche delle pipelines ETL in maniera efficiente.

L'utilizzo di Microsoft SQL Server come tipo di database è dovuto alla familiarità del tesista con gli strumenti Microsoft. Tuttavia, è importante notare che questa scelta non è limitante, poiché le istruzioni SQL utilizzate sono standard per tutti i DBMS relazionali. Inoltre, i moduli Python per l'integrazione con i vari DBMS seguono una sintassi comune, semplificando il processo di modifica nel caso in cui si voglia cambiare il tipo di DBMS in futuro.

Per connettersi ai database SQL Server e eseguire operazioni di esplorazione dei dati e popolamento delle tabelle, è stata introdotta la libreria pyodbc all’interno del codice Python. Questa libreria offre un'interfaccia standard per connettersi a una vasta gamma di database, inclusi Microsoft SQL Server, e consente di eseguire query SQL e altre operazioni di gestione del database in modo efficiente utilizzando Python.

Il progetto Python è strutturato in vari script, ciascuno dedicato a una specifica funzionalità. Alcuni script gestiscono l'esplorazione delle tabelle sorgenti per recuperare i dati, mentre altri si concentrano sul modello di intelligenza artificiale per identificare i nomi delle tabelle da creare e i parametri necessari per l'ETL. Ci sono anche script che definiscono le procedure automatizzate per l'ETL, generano le tabelle, preparano i prompt per il modello di IA e leggono i file.

3.1.2 Descrizione tecnica

Il codice sorgente è disponibile al riferimento [45].

3.1.2.1 Data exploration

Nella fase iniziale della progettazione di una pipeline ETL, è essenziale condurre uno studio approfondito delle tabelle sorgenti. Questo studio serve principalmente a identificare le variabili necessarie per le operazioni ETL e a fornirle al modello per la configurazione dei parametri. La cattura dei dati dal database viene eseguita tramite le funzioni presenti nello script “data_exp.py”.

La funzione "init" si occupa di stabilire la connessione al database SQL Server e di restituire le variabili "cursor" e "conn". Queste variabili consentono l'esecuzione delle query: "cursor" viene utilizzato per eseguire le query stesse, mentre "conn" gestisce le operazioni di connessione, quali la chiusura della stessa e i commit per le istruzioni di inserimento delle righe.

Una volta connessi al database, è necessario identificare diverse variabili:

- tutti i dati della tabella sorgente;
- le chiavi primarie;
- le chiavi esterne, incluse colonne e tabelle di riferimento;
- i nomi delle colonne e delle tabelle nel database;
- i valori delle chiavi esterne delle tabelle figlie e delle tabelle padre;
- i numeri di riga (“row_number”) nella tabella DLT.

La funzione "find_data" riceve il nome della tabella come parametro e esegue un'operazione di selezione su di essa. La funzione "find_pk" estrae i nomi e i tipi delle chiavi primarie dalla tabella sorgente utilizzando le tabelle di sistema. La funzione "find_fk" estrae le chiavi esterne, insieme alle colonne e alle tabelle padre a cui fanno riferimento, sempre utilizzando le tabelle di sistema.

La funzione "find_col" individua le colonne della tabella sorgente, mentre "find_tables" recupera i nomi di tutte le tabelle disponibili. "find_fk_values" individua tutti i valori assunti dalle chiavi esterne nella tabella sorgente e nelle tabelle padre.

Infine, la funzione "find_rownumber" calcola le row_number della tabella DLT per individuare i valori duplicati e prepararli per la rimozione nella fase successiva di OK.

Ognuna di queste funzioni utilizza "mk_dframe" per trasformare i dati memorizzati nella variabile "cursor" in dataframes, rendendoli manipolabili con Python. Questo processo coinvolge la creazione di una matrice vuota, la popolazione con i record identificati e la trasformazione in un dataframe.

3.1.2.2 Modello IA

L'implementazione del modello di IA è stata realizzata attraverso la API di OpenAI, che consente di selezionare il modello da eseguire, i prompt e i parametri necessari per adattare il comportamento del modello al contesto d'uso. Nel contesto della tesi, sono stati introdotti due parametri chiave per personalizzare e adattare le risposte alle esigenze del contesto attuale: "temperature" (temperatura) e "top_p".

Il parametro "top_p" rappresenta la soglia minima della probabilità cumulativa che un insieme di token deve superare per essere considerato nella generazione del token successivo. Ad esempio, se "top_p" è impostato a 0.6, il modello selezionerà i token in ordine decrescente di probabilità fino a quando la probabilità cumulativa raggiunge o supera il valore di 0.6. In pratica, un valore basso di "top_p" limita la scelta dei token, favorendo risposte più deterministiche.

D'altra parte, il parametro "temperature" influisce sulla varianza di probabilità tra i token e determina il grado di casualità delle risposte. Valori più elevati di temperatura conducono a risposte più creative e casuali, mentre valori più bassi favoriscono risposte più deterministe.

Nel contesto della tesi, sono stati scelti valori bassi per entrambi i parametri temperatura e "top_p". Ciò è dovuto alla necessità di ottenere risposte deterministiche e controllate in risposta ai prompt forniti al modello. Questa scelta consente di guidare il modello verso una generazione di testo più mirata e coerente con gli obiettivi specifici dell'ambito di ricerca della tesi.

La funzione “ai_run_etl” ha lo scopo di fornire un prompt al modello per ottenere le variabili necessarie per progettare un processo ETL. Il prompt è diviso in tre parti: il “pre-prompt”, che contiene istruzioni generali; la parte inserita dall'utente, che descrive la tabella di

interesse, il contesto e le richieste specifiche; e il “post-prompt”, che fornisce istruzioni sul formato della risposta, che deve essere in formato JSON per renderla leggibile in Python. Nel “post-prompt”, inoltre, vengono forniti sia due esempi di risposta per fornire un riferimento al modello, che alcuni dei dati ottenuti durante la fase di esplorazione dei dati ossia i nomi delle tabelle, le colonne delle tabelle sorgenti, le chiavi primarie e le informazioni sulle chiavi esterne.

Il messaggio di risposta del modello è strutturato in formato JSON e include le seguenti variabili:

- “tables”: questo campo indica tutte le tabelle presenti nella pipeline, comprese SRC (la tabella sorgente), STG, DLT, OK, ERROR e ODS;
- “upd_date”: che rappresenta il nome della colonna che contiene la data di caricamento del record sulla tabella sorgente. Se questa colonna non è presente, andrà utilizzato il metodo MINUS per la fase DLT e il campo verrà compilato con la stringa "NULL".
- “ok”: il quale due sottocampi:
    - “not_null”: una lista delle colonne dove è necessario effettuare il controllo di non nullità;
    - "domain": un elenco dei vincoli di dominio delle colonne. Per vincoli di dominio si intendono gli insiemi di valori accettabili dei dati, come ad esempio il fatto che il prezzo di un prodotto non può essere inferiore a 0.

Oltre alla progettazione della pipeline ETL, il modello di intelligenza artificiale interviene nella generazione delle nuove tabelle attraverso la funzione “ai_run_gentables”. In tale funzione, il modello di intelligenza artificiale assume il compito di suggerire i nomi appropriati per queste tabelle a partire dal nome della tabella sorgente. Questi nomi devono riflettere le varie fasi del processo e saranno utilizzati nello script per la creazione delle tabelle stesse. L'obiettivo è garantire una chiara e coerente organizzazione delle informazioni all'interno del sistema.

In entrambe le funzioni, è integrata la possibilità di interagire con il modello tramite una simulazione di chat. Questo permette agli utenti di visualizzare la risposta iniziale del modello e, se necessario, fornire ulteriori indicazioni per eventuali correzioni. In caso la risposta soddisfi le aspettative, basterà confermarla rispondendo con il carattere "y". Questo approccio favorisce un'interazione dinamica e flessibile con il sistema, consentendo agli utenti di intervenire attivamente nel processo e garantendo che le risposte siano allineate alle loro esigenze e aspettative.

3.1.2.3 Procedure ETL

Una volta ottenute le informazioni dal database e identificati i parametri dall'algoritmo di IA, è necessario eseguire il processo di ETL. All'interno dello script “etl.py”, si trovano le funzioni corrispondenti a ciascuna tabella della pipeline, dall'origine (SRC) fino alla destinazione (ODS).

La funzione “insert_table” viene invocata da tutte le altre funzioni. Questa funzione accetta un dataframe e una stringa contenente il nome della tabella, eseguendo quindi i comandi SQL necessari per inserire le righe e popolare la tabella.

La funzione “load_stg” recupera i dati dalla tabella SRG utilizzando la funzione “find_data”, e li carica nella tabella STG insieme al campo “JOBID_L0”.

La funzione “load_dlt_minus” esegue la fase DLT utilizzando il metodo MINUS. Inizialmente, recupera i dati dalla tabella STG, memorizza in un array i valori della colonna contenenti i JOBID, il quale viene poi ordinato. Se la dimensione dell'array è pari a 1, indica che si tratta del primo caricamento, e quindi il dataframe DLT viene creato come una copia di STG, aggiungendo la colonna “FLG_NEG” con valori pari a 0. Se la dimensione dell'array è maggiore di 1, vengono identificate le due partizioni di STG con i valori di “JOBID_L0” più grandi, corrispondenti alle due partizioni più recenti, e viene eseguito il metodo MINUS. Questo viene realizzato mediante l'uso della funzione merge di pandas, producendo due dataframe: “minus0”, contenente le righe aggiunte, e “minus1”, contenente le righe cancellate. Successivamente, avviene un confronto tra i due dataframe “minus0” e

“minus1”, individuando le righe che condividono gli stessi valori delle chiavi e mantenendo solo quelle con il valore di “JOBID_L0” maggiore.
La funzione “load_dlt”, invece, individua le righe della tabella STG con la data di caricamento maggiore rispetto alla data più recente tra i record inseriti nella tabella DLT in modo da identificare i record non ancora aggiunti, e le aggiunge a quest'ultima.
Nella funzione “load_ok” viene eseguito il filtraggio dei record attraverso quattro controlli. Il primo controllo di non nullità verifica che le chiavi e gli altri campi indicati nel campo “not_null” di outcome non siano nulli, il secondo controllo di integrità referenziale verifica che i valori delle chiavi esterne siano presenti anche tra i valori delle tabelle padri. Il terzo controllo verifica che i valori dei record rispettino i vincoli di dominio imposti e, infine, si effettua un controllo sulle row_number per individuare ed escludere i dati duplicati. I dati esclusi dalla tabella OK vengono invece inseriti nella tabella di ERROR. Per ogni riga, l’esito dei controlli di non nullità, di integrità referenziale e sulle righe duplicate viene effettuato attraverso le tre variabili “checkpk”, “checkfk” e “checkrown”.
Infine, nella fase di ODS viene eseguita un’istruzione di MERGE per aggiungere i record nuovi, eliminare quelli cancellati e aggiornare i record modificati. Gli attributi di join sono le chiavi primarie, mentre gli attributi di update sono tutti gli altri.
Oltre alle funzioni di tabella vi è presente “mk_jobid”, una funzione che crea la variabile JOBID a partire dalla funzione now della libreria datetime.

3.1.2.4 Generazione automatica delle tabelle

Per accelerare il processo ETL nel caso in cui l'utente non disponga ancora delle tabelle necessarie, è stato sviluppato uno script di generazione automatica delle tabelle. Questo script si basa sulla creazione delle query SQL di creazione delle tabelle a partire dalla composizione di singole stringhe, che definiscono i campi delle tabelle insieme a tutte le relative informazioni quali lunghezza, scala, precisione e regole di confronto per le variabili di tipo stringa, e l'identificazione dei campi e delle loro proprietà avviene a partire dai dati provenienti dalla tabella INFORMATION_SCHEMA.COLUMNS. L'esecuzione delle query avviene tramite il comando "execute".

3.1.2.5 Main

Lo script di main si occupa di realizzare il processo di generazione della pipeline ETL richiamando le funzioni da tutti gli altri script.

Il processo principale inizia con la connessione al database. Le variabili necessarie per stabilire la connessione possono essere acquisite in due modi: attraverso l'inserimento manuale dall'utente tramite input sulla console, oppure, per maggiore rapidità, leggendo da un file di configurazione appositamente creato in formato JSON, il quale contiene le variabili necessarie insieme al nome della tabella sorgente.

In seguito, l'utente ha la possibilità di avviare il processo di generazione delle nuove tabelle attraverso il metodo automatico precedentemente descritto.

Il passaggio successivo coinvolge la progettazione del processo ETL. Questo può avvenire sia attraverso il file di configurazione che utilizzando un modello di intelligenza artificiale. L'utente fornisce un prompt al modello AI, che genera i parametri da mostrare all'utente.

Una volta confermati dall'utente, avviene il processo ETL effettivo. La fase STG viene eseguita prima, seguita dallo svuotamento della tabella OK. Successivamente, se il campo “upd_date” della variabile “outcome” non è vuoto, viene eseguita la procedura di eliminazione standard (DLT); altrimenti, viene utilizzato il metodo “MINUS”. Infine, vengono eseguite le funzioni per le fasi OK e ODS.

3.2 Generazione automatica del progetto SSIS tramite Python

3.2.1 Descrizione generale e scelte progettuali

Successivamente al primo progetto, il tesista si è affiancato a un altro gruppo di lavoro per supportare la realizzazione di un secondo progetto che si pone lo stesso scopo intraprendendo, tuttavia, una strada completamente diversa.

Mentre il primo progetto utilizza degli script Python per eseguire le query SQL che costituiscono il processo di ETL, il secondo gruppo di lavoro ha invece utilizzato Python per realizzare il processo di generazione automatica di files XML, dai quali la piattaforma SSIS genera i pacchetti necessari per realizzare la pipeline ETL.

Nel contesto di questo progetto il compito del modello di intelligenza artificiale, anche in questo caso GPT-3.5, consiste nella capacità di selezionare le funzioni specifiche da eseguire. Queste funzioni corrispondono a pacchetti distinti e, di conseguenza, a fasi diverse del processo ETL. Il modello agisce come un chatbot, consentendo all'utente di eseguire o progettare le parti del processo che desidera utilizzando il linguaggio naturale.

3.2.2 Descrizione tecnica

Per lo sviluppo del processo di generazione automatico del file XML, sono stati realizzati diversi script a cui corrispondono le parti del codice XML.
Gli script consistono nella composizione di una stringa che viene poi formattata in file XML.
Il tesista, nell'ambito del nuovo progetto, si è occupato di sviluppare alcune parti secondarie e accessorie.

3.2.2.1 Composizione del progetto SSIS

Il flusso di lavoro SSIS implementato nella tesi è suddiviso in tre pacchetti principali. Il primo pacchetto, denominato "Main", funge da orchestratore globale. Nello specifico, esso si occupa delle operazioni descritte di seguito.
Il processo inizia con la creazione del "JOBID", un identificativo univoco per il processo in esecuzione. Successivamente, il pacchetto esegue gli altri due pacchetti e avvia l'esecuzione del processo di ETL. Prima di iniziare, il pacchetto esegue anche un comando truncate sulla tabella di destinazione "OK", assicurandosi che sia pronta per un nuovo caricamento.
Il secondo pacchetto, denominato "L0", rappresenta il livello L0 del processo ETL. Questo pacchetto comprende due attività principali di flusso dati: STG e DLT. Nel flusso STG, i dati vengono estratti dalla tabella SRC, viene creata la colonna derivata JOBID e successivamente i dati vengono caricati nella tabella STG.
Per quanto riguarda la fase DLT, vengono eseguiti due caricamenti distinti. Il primo utilizza un'operazione di minus tra i dati caricati su STG nell'ultimo caricamento e quelli del penultimo, chiamato "OGGI-IERI". Questo serve a individuare le righe aggiunte rispetto all'ultimo caricamento. Il secondo caricamento esegue un'operazione di minus tra il penultimo e l'ultimo, chiamato "IERI-OGGI", individuando le righe che sono state cancellate.

Per le righe provenienti da "OGGI-IERI", il campo “FLG_NEG” viene impostato a 0, mentre per "IERI-OGGI" viene impostato a 1, indicando che la riga è stata cancellata. I due flussi vengono poi uniti e i dati vengono caricati nella tabella DLT.

Il terzo pacchetto, corrispondente al livello L1, è composto da due attività principali: una di tipo flusso dati denominata OK e l’altra di tipo esegui SQL denominata ODS. Nell'attività di OK, vengono eseguiti controlli di non nullità, integrità referenziale, di dominio e di “row_number”, per escludere i duplicati. Le righe che superano tutti i controlli vengono instradate verso l'output "OK" e successivamente caricate nella tabella OK. Le righe che non superano i controlli vengono instradate verso l'output "E$" e caricate nella tabella degli errori. Infine, nell'attività di ODS, viene eseguita la query di merge finale tra ODS e OK, consolidando i dati e completando il processo di ETL.

3.2.2.2 Composizione del file XML

Il file XML utilizzato da SSIS per generare il pacchetto è organizzato in una struttura gerarchica che riflette la composizione complessiva delle attività e dei flussi dati. Ogni pacchetto è rappresentato dal tag "package". All'interno di questo tag, si trovano gli elementi del flusso di controllo e dati chiamati "dts executable".

Questi "dts executable" contengono a loro volta gli elementi del flusso di dati denominati "component". Ogni componente ha i tag "input" e "output", che descrivono le colonne associate a ciascun flusso. Le colonne sono specificate nei tag "inputcolumns" e "outputcolumns" e, per ciascuna colonna, vengono definiti i tag "inputcolumn" e "outputcolumn".

Il tag "connection" viene utilizzato per configurare le connessioni ai database necessarie per eseguire le attività presenti nel pacchetto. I collegamenti tra gli elementi del flusso di dati e tra le attività sono indicati dal tag "paths".

Per specificare proprietà aggiuntive di colonne e di elementi del flusso dati si utilizza il tag "property". Questo tag consente di definire attributi specifici per personalizzare ulteriormente il comportamento del pacchetto SSIS.

3.2.2.3 Generazione del file Excel di configurazione

I dati da inserire nella compilazione del file XML sono presenti all'interno di una tabella Excel di configurazione a cui lo script fa riferimento. Data la grandezza del file Excel, è necessario che la generazione di esso sia automatica per ridurre significativamente i tempi. All'interno del file Excel sono presenti tutti i dati relativi alle colonne per compilare il file XML ossia:

- i nomi;
- il flag denominato "key", che indica se la colonna è chiave primaria della tabella sorgente. Se sì, il flag è impostato su "yes", altrimenti è vuoto;
- il tipo di variabile da utilizzare per il campo in SSIS, considerando che la nomenclatura differisce dallo standard SQL. Ad esempio, la stringa Unicode viene indicata in SQL con "NVARCHAR", mentre in SSIS con "wstr";
- il nome della componente del flusso di dati da cui il componente STG ricava la colonna;
- il tipo di variabile da indicare nell'operatore "Colonna derivata";
- informazioni sulla lunghezza, precisione e scala delle variabili, a seconda che siano stringhe o numerici;
- gli output dai quali vengono ottenuti i valori delle colonne nelle diverse fasi;
- le espressioni da utilizzare negli operatori "Colonna derivata" durante la fase DLT.

Per ottenere i dati necessari alla configurazione del file Excel, è stato creato un dataframe con nome "dfcols" contenente tutte le informazioni dalla tabella INFORMATION_SCHEMA.COLUMNS. Tale operazione è contenuta nella funzione "gen_dfcols". Questo dataframe include i nomi dei campi della tabella sorgente, le tipologie di dati e i relativi parametri come lunghezza, precisione e scala.

Successivamente, per la colonna "keys" viene eseguita una seconda query sulle tabelle INFORMATION_SCHEMA.KEY_COLUMN_USAGE e INFORMATION_SCHEMA.TABLE_CONSTRAINTS per identificare i nomi dei campi che

fungono da chiavi primarie. Viene quindi eseguito un confronto del risultato con le colonne presenti nel dataframe per compilare la colonna.
Per effettuare le conversioni dei nomi dei tipi di variabili da SQL a SSIS, si è utilizzata una variabile dizionario chiamata “sql2ssis”. Quest'ultima serve a mappare i tipi di dati SQL ai corrispondenti tipi di dati SSIS.
Il resto dei campi del dataframe viene compilato utilizzando stringhe predefinite, poiché la struttura dei pacchetti SSIS e la loro nomenclatura sono definite in anticipo.
Per ricavare il file Excel viene eseguita la funzione di Pandas “to_excel” a partire dal dataframe “excel”.

3.3 Confronto tra i due progetti

Sebbene i due progetti realizzati abbiano lo stesso scopo, ossia quello di generare automaticamente una pipeline ETL con l’ausilio di un modello di IA generativa, essi utilizzano due modalità completamente differenti, e il progetto che prevede la generazione automatica del pacchetto SSIS ha diversi vantaggi sulla pipeline generata interamente in Python.
Il primo punto di forza del secondo progetto è la semplicità di configurazione del pacchetto SSIS.
Il primo progetto ha a disposizione la possibilità di utilizzare due file di configurazione per automatizzare le operazioni relative alla connessione al database e alla progettazione del processo ETL, qualora non si volessero utilizzare le funzionalità di input tramite console. Tuttavia, le possibilità di personalizzazione del processo accessibili agli utenti meno esperti sono limitate, mentre le restanti richiedono la modifica diretta del codice. Ciò comporta un aumento significativo dei tempi in caso di necessità di personalizzazione del processo dovuto alla complessità e ai tempi di sviluppo.
Il secondo progetto, al contrario, esegue le operazioni di generazione della pipeline ETL attraverso un file Excel di configurazione che contiene tutti i parametri necessari. La personalizzazione del processo a proprio piacimento richiede quindi una conoscenza pregressa del formato della tabella Excel che, se confrontato con la conoscenza di un

linguaggio di programmazione, risulta notevolmente più facile e immediato. Qualora ciò non fosse possibile, è necessario modificare i pacchetti SSIS successivamente alla loro generazione. La piattaforma SSIS include un'interfaccia grafica che rappresenta in modo intuitivo le sequenze di operazioni e i flussi di dati, facilitando ed espandendo ulteriormente le possibilità di configurazione della propria pipeline.

Oltre a ciò, il funzionamento del modello di intelligenza artificiale è completamente diverso nei due progetti. Nel primo progetto il compito del modello è quello di suggerire i nomi delle nuove tabelle, qualora si dovessero generare, e di suggerire i parametri e le modalità di esecuzione del processo ETL, che è eseguito nella sua interezza.

Al contrario, nel secondo progetto il compito del modello di IA consiste nel comprendere quali parti del processo ETL eseguire o progettare.

In sintesi, il progetto di generazione automatica dei pacchetti SSIS consente una maggiore personalizzazione sia per quanto riguarda la progettazione della pipeline ETL sia per le esecuzioni.

Inoltre, la piattaforma SSIS è proprietaria di Microsoft ed è consolidata sul mercato, ha costi accessibili e prevede un servizio di supporto, mentre uno script Python non ha niente di tutto ciò.

4 Risultati e validazione

All'interno di questo capitolo, si condurrà un'analisi dei risultati derivanti dall'impiego del primo tool realizzato. Per la parte di test sono stati impiegati dei dataset forniti dall'azienda Mediamente Consulting srl, che rappresentano dei dati relativi alle vendite. A ogni dataset è stato assegnato un database apposito e una tabella sorgente, sebbene non fosse necessario e si potessero accorpare tutte le tabelle sorgenti in un unico database. Nell'impostazione delle connessioni si è deciso di utilizzare sempre la connessione Windows; se si vuole utilizzare un utente specifico basterà garantirgli i permessi di creazione delle tabelle, di inserimento all'interno di esse e di accesso alle tabelle sorgenti. Per il test, è stato anche utilizzato un dataset presente sulla piattaforma Kaggle contenente i dati relativi alle vendite di una catena di supermercati.

Di seguito, sono riportate le strutture delle tabelle sorgenti:

Nome colonna	Tipo di dati	Consenti valori Null
Invoice_ID	varchar(100)	☐
Branch	char(1)	☑
City	varchar(100)	☑
Customer_type	varchar(100)	☑
Gender	varchar(100)	☑
Product_line	varchar(100)	☑
Unit_price	decimal(10, 2)	☑
Quantity	int	☑
Tax_5	decimal(10, 2)	☑
Total	decimal(10, 2)	☑
Date	date	☑
Time	time(7)	☑
Payment	varchar(100)	☑
cogs	decimal(10, 2)	☑
gross_margin_percentage	decimal(5, 2)	☑
gross_income	decimal(10, 2)	☑
Rating	decimal(3, 1)	☑
		☐

FIGURA 6. COLONNE DELLA TABELLA SUPERMKT_SALES_SRC

	Nome colonna	Tipo di dati	Consenti valori Null
🔑	Division	varchar(255)	☐
🔑	Store	varchar(255)	☐
🔑	Week	varchar(255)	☐
	[WP Sales LC V]	decimal(10, 2)	☑
	[PL Sales LC V]	decimal(10, 2)	☑
	[PL Discount LC V]	decimal(10, 2)	☑
	[WP Initial Margin V]	decimal(10, 2)	☑
	[LF Sales LC V]	decimal(10, 2)	☑
	[WP Discount LC V]	decimal(10, 2)	☑
	[LF Discount LC V]	decimal(10, 2)	☑
	Timestamp	datetime2(7)	☑
			☐

FIGURA 7. COLONNE DELLA TABELLA SRC_DIV_STORE_WK

Nome colonna	Tipo di dati	Consenti valori Null
Subcategory	varchar(25)	☐
[Country Channel]	varchar(25)	☐
Seasonality	varchar(25)	☐
Week	varchar(25)	☐
[PL Sales V]	decimal(18, 2)	☑
[PL Sales U]	decimal(18, 2)	☑
[PL Discount V]	decimal(18, 2)	☑
[PL Stock Target V]	decimal(18, 2)	☑
[PL OTB V]	decimal(18, 2)	☑
[PL OTB U]	decimal(18, 2)	☑
[PL Stock Close V]	decimal(18, 2)	☑
[PL Stock Close U]	decimal(18, 2)	☑
[WP Stock Close Fwd Co...	decimal(18, 2)	☑
[PL Stock Close Fwd Cov...	decimal(18, 2)	☑
[WP SALES V]	decimal(18, 2)	☑
[LF SALES V]	decimal(18, 2)	☑
[WP Discount V]	decimal(18, 2)	☑
[LF Discount V]	decimal(18, 2)	☑
[WP Proposed Orders V]	decimal(18, 2)	☑
[WP Proposed Orders U]	decimal(18, 2)	☑
[WP OTB V - ro]	decimal(18, 2)	☑
[WP OTB U - ro]	decimal(18, 2)	☑
[WP Stock Close V]	decimal(18, 2)	☑
[WP Stock Close U]	decimal(18, 2)	☑
[LF Proposed Orders V]	decimal(18, 2)	☑
[LF Proposed Orders U]	decimal(18, 2)	☑
[LF OTB V - ro]	decimal(18, 2)	☑
[LF OTB U - ro]	decimal(18, 2)	☑
[LF Stock Close V]	decimal(18, 2)	☑
[LF Stock Close U]	decimal(18, 2)	☑
Timestamp	datetime2(7)	☐
		☐

FIGURA 8. COLONNE DELLA TABELLA SUBCAT_CC_SEAS_WK

Nome colonna	Tipo di dati	Consenti valori Null
Subcategory	varchar(25)	☐
Store	varchar(25)	☐
Seasonality	varchar(25)	☐
[WP Sales V MG]	decimal(18, 2)	☑
[PL Sales V MG]	decimal(18, 2)	☑
[WP Discount POS V MG]	decimal(18, 2)	☑
[PL Discount POS V MG]	decimal(18, 2)	☑
[WP Buy U]	decimal(18, 2)	☑
[WP Buy V]	decimal(18, 2)	☑
[WP Buy V Inc Vat]	decimal(18, 2)	☑
Timestamp	datetime2(7)	☐

FIGURA 9. COLONNE DELLA TABELLA SRC_SUBCAT_STORE_SEAS

Nome colonna	Tipo di dati	Consenti valori Null
Division	varchar(255)	☐
[Country Channel]	varchar(255)	☐
Month	varchar(255)	☐
[PL Sales V]	decimal(20, 2)	☑
[PL Sales Margin V]	decimal(20, 2)	☑
Timestamp	datetime2(7)	☑
		☐

FIGURA 10. COLONNE DELLA TABELLA SRC_DIV_CC_MN

Di seguito sono riportati i risultati dei due prompt di generazione delle tabelle e di progettazione del processo ETL per le tabelle.

Risultati per la tabella “SRC_DIV_CC_MN”:

```
{
  "stg": "STG_DIV_CC_MN",
  "dlt": "DLT_DIV_CC_MN",
  "ok": "OK_DIV_CC_MN",
  "e$": "E$_DIV_CC_MN",
  "ods": "ODS_DIV_CC_MN"
}
```

```
{
    "tables":
    {
      "src": "SRC_DIV_CC_MN",
      "stg": "STG_DIV_CC_MN",
      "dlt": "DLT_DIV_CC_MN",
      "ok": "OK_DIV_CC_MN",
      "error": "E$_DIV_CC_MN",
      "ods": "ODS_DIV_CC_MN"
    },
    "upd_date": "Timestamp",
    "ok":
    {
      "not_null": ["Division", "Country Channel", "Month", "PL Sales
V", "PL Sales Margin V"],
      "domain": "NULL"
    }
}
```

Risultati per la tabella “src_div_store_wk”:

```
{
  "stg": "STG_DIV_STORE_WK",
  "dlt": "DLT_DIV_STORE_WK",
  "ok": "OK_DIV_STORE_WK",
  "e$": "E$_DIV_STORE_WK",
  "ods": "ODS_DIV_STORE_WK"
}

{
    "tables":
    {
      "src": "src_div_store_wk",
      "stg": "STG_DIV_STORE_WK",
      "dlt": "DLT_DIV_STORE_WK",
      "ok": "OK_DIV_STORE_WK",
      "error": "E$_DIV_STORE_WK",
      "ods": "ODS_DIV_STORE_WK"
    },
    "upd_date": "Timestamp",
    "ok":
    {
```

```
      "not_null": ["Division", "Store", "Week", "WP Sales LC V", "PL
Sales LC V", "PL Discount LC V", "WP Initial Margin V", "LF Sales
LC V", "WP Discount LC V", "LF Discount LC V"],
      "domain": "NULL"
    }
}
```

Risultati per la tabella “src_subcat_cc_seas_wk”:

```
{
  "stg": "STG_subcat_cc_seas_wk",
  "dlt": "DLT_subcat_cc_seas_wk",
  "ok": "OK_subcat_cc_seas_wk",
  "e$": "E$_subcat_cc_seas_wk",
  "ods": "ODS_subcat_cc_seas_wk"
}

{
    "tables":
    {
      "src": "src_subcat_cc_seas_wk",
      "stg": "STG_SUBCAT_CC_SEAS_WK",
      "dlt": "DLT_SUBCAT_CC_SEAS_WK",
      "ok": "OK_SUBCAT_CC_SEAS_WK",
      "error": "E$_SUBCAT_CC_SEAS_WK",
      "ods": "ODS_SUBCAT_CC_SEAS_WK"
    },
    "upd_date": "Timestamp",
    "ok":
    {
      "not_null": ["Subcategory", "Country Channel", "Seasonality",
"Week", "PL Sales V", "PL Sales U", "PL Discount V", "PL Stock Target
V", "PL OTB V", "PL OTB U", "PL Stock Close V", "PL Stock Close U",
"WP Stock Close Fwd Cover", "PL Stock Close Fwd Cover", "WP SALES
V", "LF SALES V", "WP Discount V", "LF Discount V", "WP Proposed
Orders V", "WP Proposed Orders U", "WP OTB V - ro", "WP OTB U - ro",
"WP Stock Close V", "WP Stock Close U", "LF Proposed Orders V", "LF
Proposed Orders U", "LF OTB V - ro", "LF OTB U - ro", "LF Stock Close
V", "LF Stock Close U"],
      "domain": "NULL"
    }
  }
```

Risultati per la tabella “SRC_SUBCAT_STORE_SEAS”:

```
{
  "stg": "STG_SUBCAT_STORE_SEAS",
  "dlt": "DLT_SUBCAT_STORE_SEAS",
  "ok": "OK_SUBCAT_STORE_SEAS",
  "e$": "E$_SUBCAT_STORE_SEAS",
  "ods": "ODS_SUBCAT_STORE_SEAS"
}
```

```
{
    "tables":
    {
      "src": "src_subcat_store_seas",
      "stg": "STG_SUBCAT_STORE_SEAS",
      "dlt": "DLT_SUBCAT_STORE_SEAS",
      "ok": "OK_SUBCAT_STORE_SEAS",
      "error": "E$_SUBCAT_STORE_SEAS",
      "ods": "ODS_SUBCAT_STORE_SEAS"
    },
    "upd_date": "Timestamp",
    "ok":
    {
      "not_null": ["Subcategory", "Store", "Seasonality", "WP Sales
V MG", "PL Sales V MG", "WP Discount POS V MG", "PL Discount POS V
MG", "WP Buy U", "WP Buy V", "WP Buy V Inc Vat"],
      "domain":
      {
        "WP Sales V MG": "> 0",
        "PL Sales V MG": "> 0",
        "WP Discount POS V MG": ">= 0",
        "PL Discount POS V MG": ">= 0",
        "WP Buy U": ">= 0",
        "WP Buy V": ">= 0",
        "WP Buy V Inc Vat": ">= 0"
      }
    }
}
```

Risultati per la tabella “SUPERMKT_SALES_SRC”. Per questa tabella, sono state fatte delle prove per valutare la capacità del modello di effettuare le correzioni richieste dall’utente:

```
{
    "tables":
    {
      "stg": "SUPERMKT_SALES_STG",
      "dlt": "SUPERMKT_SALES_DLT",
      "ok": "SUPERMKT_SALES_OK",
      "error": "SUPERMKT_SALES_E$",
      "ods": "SUPERMKT_SALES_ODS"
    },
    "upd_date": "NULL",
    "ok":
    {
      "not_null": ["Invoice_ID", "Branch", "City", "Customer_type",
"Gender",   "Product_line",   "Unit_price",   "Quantity",   "Tax_5",
"Total",       "Date",       "Time",       "Payment",       "cogs",
"gross_margin_percentage", "gross_income", "Rating"],
      "domain": "NULL"
    }
}
Inserire aggiustamenti aggiuntivi, per confermare inserire y:
quantità maggiori di 0
{
    "tables":
    {
      "src": "SUPERMKT_SALES_SRC",
      "stg": "SUPERMKT_SALES_STG",
      "dlt": "SUPERMKT_SALES_DLT",
      "ok": "SUPERMKT_SALES_OK",
      "error": "SUPERMKT_SALES_E$",
      "ods": "SUPERMKT_SALES_ODS"
    },
    "upd_date": "NULL",
    "ok":
    {
      "not_null": ["Invoice_ID", "Branch", "City", "Customer_type",
"Gender",   "Product_line",   "Unit_price",   "Quantity",   "Tax_5",
"Total",       "Date",       "Time",       "Payment",       "cogs",
"gross_margin_percentage", "gross_income", "Rating"],
      "domain":
      {
        "Quantity": "> 0"
```

```
      }
    }
}
Inserire aggiustamenti aggiuntivi, per confermare inserire y: costo
del venduto maggiore di 0
{
    "tables":
    {
      "src": "SUPERMKT_SALES_SRC",
      "stg": "SUPERMKT_SALES_STG",
      "dlt": "SUPERMKT_SALES_DLT",
      "ok": "SUPERMKT_SALES_OK",
      "error": "SUPERMKT_SALES_E$",
      "ods": "SUPERMKT_SALES_ODS"
    },
    "upd_date": "NULL",
    "ok":
    {
      "not_null": ["Invoice_ID", "Branch", "City", "Customer_type",
"Gender",   "Product_line",   "Unit_price",   "Quantity",   "Tax_5",
"Total",       "Date",       "Time",       "Payment",       "cogs",
"gross_margin_percentage", "gross_income", "Rating"],
      "domain":
      {
        "Quantity": "> 0",
        "cogs": "> 0"
      }
    }
}
```

Dai risultati si evince che il modello è stato capace di suggerire dei nomi efficaci per le nuove tabelle, facendo in modo che seguano la stessa forma del nome della tabella sorgente. Per quanto riguarda la progettazione delle pipeline, il modello è stato capace di assegnare in maniera corretta le tabelle alle singole fasi, tuttavia si evincono tre problemi.

Il primo di essi è l'assegnazione del campo Timestamp come data di update per la fase DLT, che è incorretto. Ciò è dovuto al fatto che il termine Timestamp può generalmente causare confusione persino a un progettista umano, se non viene fornito il giusto contesto. Una volta

esplicato che Timestamp non rappresenta la data di aggiornamento, il modello ha assegnato la stringa NULL, ossia la risposta corretta, al campo “upd_date”.
Il secondo problema riguarda l’assegnazione delle colonne che non devono essere nulle, che il modello assegna sempre a tutti i campi, a eccezione di Timestamp. Ciò può essere un problema in quanto risulterebbe un vincolo troppo stringente.
Infine, tra tutti i test il modello ha assegnato dei vincoli di dominio soltanto una volta, quando tutte le tabelle hanno dei campi rappresentanti le vendite che devono essere maggiori o uguali di 0.
Questi tre problemi sono probabilmente dovuti a diversi fattori. Il primo è la qualità dei prompt, dovuto all’inesperienza del tesista. Il secondo è il fatto che il tesista, nel processo di progettazione della pipeline, non ha fornito il contesto riguardante la tabella e i campi, informazioni che, inserite successivamente, hanno portato il modello ad aggiustare le risposte.
Nonostante ciò va fatto notare che, di fronte a richieste in linguaggio naturale, il modello è capace di risolvere i tre problemi citati in precedenza e in relativamente poco tempo, portando a un risultato nel complesso positivo.

5 Conclusione e sviluppi futuri

Il progetto attuale offre molteplici opportunità per lo sviluppo futuro. Una delle direzioni chiave è la progettazione completa del processo ETL. Attualmente, i due strumenti automatizzano il processo ETL dalla tabella STG alla tabella ODS. Tuttavia, si prevede di estendere il supporto fino al livello L2, includendo le tabelle PUB e L2. Queste fasi sono più complesse e richiedono un'analisi dettagliata del contesto e delle esigenze del cliente.

Una direzione cruciale per lo sviluppo dell'implementazione riguarda l'introduzione della capacità di gestire dati non strutturati. Attualmente, il tool dispone solo delle funzionalità per trattare dati già strutturati in tabelle SQL. L'implementazione di questa funzionalità consentirebbe l'analisi di una grande quantità di dati altrimenti non accessibili. L'integrazione della gestione dei dati non strutturati aprirebbe le porte a una vasta gamma di informazioni provenienti da fonti come testi, documenti, immagini e file multimediali. Questo amplierebbe notevolmente il campo delle informazioni analizzabili, consentendo una comprensione più approfondita dei dati.

È, altresi', necessario migliorare le performance del modulo per quanto riguarda la progettazione dei processi ETL. Sebbene al momento le correzioni non richiedano molto tempo, esiste spazio per ideare prompt migliori e più chiari per ottimizzare l'efficienza complessiva.

Inoltre, si potrebbe considerare l'implementazione del modulo intelligenza artificiale a livello ancora più astratto per personalizzare le colonne e i metodi utilizzati nei flussi di dati. Poiché i metodi di realizzazione dei processi ETL possono variare tra le organizzazioni, la sfida futura sarà consentire una maggiore personalizzazione in questo ambito.

Un'altra prospettiva di sviluppo riguarda il supporto per un numero più ampio di piattaforme. Attualmente, i tool sono progettati per lavorare con database SQL Server di Microsoft, ma considerare la possibilità di estendere il supporto a database come Oracle, mySQL e SQLite aumenterebbe la flessibilità del sistema.

Per quanto riguarda il lungo termine, potrebbe essere vantaggioso implementare un modello di intelligenza artificiale a livello locale in azienda per ridurre i costi associati

all'utilizzo di servizi di terze parti. Considerando l'evoluzione continua dei modelli open source di intelligenza artificiale, potrebbe essere necessario aggiornare i tool per integrare nuovi modelli più performanti o specificamente addestrati per la gestione dei processi ETL. Fare ciò richiede un processo di aggiornamento del codice e la reingegnerizzazione dei prompt utilizzati.

In conclusione, nonostante i progetti realizzati siano suscettibili di miglioramenti significativi, fin dalle prime fasi dimostrano alcuni dei principali vantaggi derivanti dall'utilizzo degli algoritmi di intelligenza artificiale per ottimizzare la progettazione dei processi ETL.

Bibliografia

[1] Almeida, João & Pazos, Alejandro & Oliveira, José. (2022). “BIcenter-AD: Harmonising Alzheimer’s Disease cohorts using a common ETL tool.” *Informatics in Medicine Unlocked*. 35. 101133. 10.1016/j.imu.2022.101133.

[2] K. V. Phanikanth and S. D. Sudarsan, "A big data perspective of current ETL techniques," *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Durban, South Africa, 2016, pp. 330-334, doi: 10.1109/ICACCE.2016.8073770.

[3] A. Sabtu *et al.*, "The challenges of Extract, Transform and Loading (ETL) system implementation for near real-time environment," *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*, Langkawi, Malaysia, 2017, pp. 1-5, doi: 10.1109/ICRIIS.2017.8002467.

[4] A. Suleykin and P. Panfilov, "Metadata-Driven Industrial-Grade ETL System," *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, 2020, pp. 2433-2442, doi: 10.1109/BigData50022.2020.9378367.

[5] Xavier, Cristiano & Moreira, Fernando. (2013). “Agile ETL.” *Procedia Technology*. 9. 381-387. 10.1016/j.protcy.2013.12.042.

[6] S. -S. Kim, W. -R. Lee and J. -H. Go, "A Study on Utilization of Spatial Information in Heterogeneous System Based on Apache NiFi," *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), 2019, pp. 1117-1119, doi: 10.1109/ICTC46691.2019.8939734.

[7] J. Sreemathy, R. Brindha, M. Selva Nagalakshmi, N. Suvekha, N. Karthick Ragul and M. Praveennandha, "Overview of ETL Tools and Talend-Data Integration," *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2021, pp. 1650-1654, doi: 10.1109/ICACCS51430.2021.9441984.

[8] O. V. Sawant, "Combating Dirty Data using Data Virtualization," *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, Bombay, India, 2019, pp. 1-5, doi: 10.1109/I2CT45611.2019.9033690.

[9] A. H. Mousa and N. Shiratuddin, "Data Warehouse and Data Virtualization Comparative Study," *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, Dubai, United Arab Emirates, 2015, pp. 369-372, doi: 10.1109/DeSE.2015.26.

[10] Y. Peng, X. Pan, S. Wang, C. Wang, J. Wang and J. Wu, "An Aero-Engine RUL Prediction Method Based on VAE-GAN," *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Dalian, China, 2021, pp. 953-957, doi: 10.1109/CSCWD49262.2021.9437836.

[11] M. Vubangsi, S. U. Abidemi, O. Akanni, A. S. Mubarak and F. Al-Turjman, "Applications of Transformer Attention Mechanisms in Information Security: Current Trends and Prospects," *2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs)*, Nicosia, Cyprus, 2022, pp. 101-105, doi: 10.1109/AIoTCs58181.2022.00021.

[12] S. Imai, "Is GitHub Copilot a Substitute for Human Pair-programming? An Empirical Study," *2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, Pittsburgh, PA, USA, 2022, pp. 319-321, doi: 10.1145/3510454.3522684.

[13] C. Rubio, F. Mella, C. Martínez, A. Segura and C. Vidal, "Exploring Copilot Github to Automatically Solve Programming Problems in Computer Science Courses," *2023 42nd IEEE International Conference of the Chilean Computer Science Society (SCCC)*, Concepcion, Chile, 2023, pp. 1-8, doi: 10.1109/SCCC59417.2023.10315758.

[14] Y. Tang, X. Dai and Y. Lv, "ChatGPT Participates in Traffic Control as a Traffic Manager Assistant," *2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, Orlando, FL, USA, 2023, pp. 1-6, doi: 10.1109/DTPI59677.2023.10365318.

[15] D. Yoo, D. Y. J. Kim and E. Lopes, "Digital Art Therapy with Gen AI: Mind Palette," *2023 11th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, Cambridge, MA, USA, 2023, pp. 1-3, doi: 10.1109/ACIIW59127.2023.10388174.

[16] Introzzi, Luca & Zonca, Joshua & Cabitza, Federico & Cherubini, Paolo & Reverberi, Carlo. (2023). “Enhancing human-AI collaboration: The case of colonoscopy.” *Digestive and Liver Disease*. 10.1016/j.dld.2023.10.018.

[17] Gryncewicz, Wieslawa & Zygala, Ryszard & Pilch, Agnieszka. (2023). “AI in HRM: case study analysis. Preliminary research.” *Procedia Computer Science*. 225. 2351-2360. 10.1016/j.procs.2023.10.226.

[18] Hassan, Mohammed Salah & Al Halbusi, Hussam & Abdelfattah, Fadi. (2023). “May the force of text data analysis be with you: Unleashing the power of generative AI for social psychology research.” *Computers in Human Behavior: Artificial Humans*. 1. 100006. 10.1016/j.chbah.2023.100006.

[19] Mohamed, Azmi & Mansour, Abdeljebar & Azmi, Chaimaa. (2023). “A Context-Aware Empowering Business with AI: Case of Chatbots in Business Intelligence Systems.” *Procedia Computer Science*. 224. 479-484. 10.1016/j.procs.2023.09.068.

[20] O. A. Tcukanova, A. A. Yarskaya and A. A. Torosyan, "Artificial Intelligence as a New Stage in the Development of Business Intelligence Systems," *2022 International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*, Saint Petersburg, Russian Federation, 2022, pp. 315-318, doi: 10.1109/ITQMIS56172.2022.9976832.

[21] Uncovering the dark side of AI-based decision-making: A case study in a B2B context

[22] M. A. Junaid, S. Anwar, G. Sikander and M. T. Khan, "Generative Adversarial Network based Chest Disease Detection and Binary Mask Generation," *2023 International Conference on Robotics and Automation in Industry (ICRAI)*, Peshawar, Pakistan, 2023, pp. 1-7, doi: 10.1109/ICRAI57502.2023.10089542.

[23 M. R. Shoaib, Z. Wang, M. T. Ahvanooey and J. Zhao, "Deepfakes, Misinformation, and Disinformation in the Era of Frontier AI, Generative AI, and Large AI Models," *2023 International Conference on Computer and Applications (ICCA)*, Cairo, Egypt, 2023, pp. 1-7, doi: 10.1109/ICCA59364.2023.10401723.

[24] P. S. Diouf, A. Boly and S. Ndiaye, "Variety of data in the ETL processes in the cloud: State of the art," *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, Bangkok, Thailand, 2018, pp. 1-5, doi: 10.1109/ICIRD.2018.8376308.

[44] Stefania Giannini, 2023, Reflections on generative AI and the future of education. © UNESCO 2023

Sitografia

[25] Preetipadma Khandavilli, Hevo Data, “Importance of ETL: 3 Critical Benefits and Top ETL Tools”, 25/02/2023, https://hevodata.com/learn/importance-of-etl/.

[26] ORACLE, “Descrizione di ETL”, https://www.oracle.com/it/integration/what-is-etl/

[27] Genisys Group, “Emerging Trends of ETL- Big Data And Beyond”, https://genisys-group.com/blog/emerging-trends-of-etl-big-data-and-beyond/.

[28] Wikipedia, “Data virtualization”, https://en.wikipedia.org/wiki/Data_virtualization.

[29] Adam Zewe, MIT News, “Explained: Generative AI”, 9/11/2023, https://news.mit.edu/2023/explained-generative-ai-1109.

[30] Wikipedia, “Data virtualization”, https://it.wikipedia.org/wiki/Proprietà_di_Markov.

[31] Wikipedia, “Rete generativa avversaria”, https://it.wikipedia.org/wiki/Rete_generativa_avversaria.

[32] Wikipedia, “Diffusion model”, https://en.wikipedia.org/wiki/Diffusion_model.

[33] Wikipedia, “Word embedding”, https://it.wikipedia.org/wiki/Word_embedding.

[34] Hiren Dhaduk, “How Does Generative AI Work: A Deep Dive into Generative AI Models”, 23/05/2023 https://www.simform.com/blog/how-does-generative-ai-work/.

[35] Cem Dilmegani, “Top 100+ Generative AI Applications / Use Cases in 2024”, 20/02/2023, https://research.aimultiple.com/generative-ai-applications/.

[36] Michael Rumiantsau, Narrative BI, “Generative BI: Setting a New Standard for Business Intelligence”, 16/11/2023, https://www.narrative.bi/analytics/generative-bi.

[37] John Burke, TechTarget, “What are the risks and limitations of generative AI?”, 13/11/2023, https://www.techtarget.com/searchEnterpriseAI/tip/What-are-the-risks-and-limitations-of-generative-AI.

[38] Bernard Marr, “The 10 Biggest Generative AI Trends For 2024 Everyone Must Be Ready For Now”, 2/10/2023, https://www.forbes.com/sites/bernardmarr/2023/10/02/the-10-biggest-generative-ai-trends-for-2024-everyone-must-be-ready-for-now/.

[39] Melissa Heikkilä, Will Douglas Heaven, “What’s next for AI in 2024”, 4/01/2024, https://www.technologyreview.com/2024/01/04/1086046/whats-next-for-ai-in-2024/.

[40] I. Glenn Cohen, Theodoros Evgeniou, Martin Husovec, “Navigating the New Risks and Regulatory Challenges of GenAI”, 20/11/2023, https://hbr.org/2023/11/navigating-the-new-risks-and-regulatory-challenges-of-genai.

[41] Apoorva Verma, “The Future of Extract, Transform & Load (ETL) Tools with NewFangled: A No-Code Generative AI-Driven Revolution”, 2/01/2024, https://newfangled.io/blog/future-of-etl-tools-with-newfangled-genai/.

[42] Tobias Macey, Jay Mishra, “Building ETL Pipelines With Generative AI”, 21/10/2023, https://www.dataengineeringpodcast.com/building-etl-pipelines-with-generative-ai-episodde-394.

[43] Cloud Data Insights, “Unlocking Autonomous Data Pipelines with Generative AI”, https://www.clouddatainsights.com/unlocking-autonomous-data-pipelines-with-generative-ai/.

[45] https://drive.google.com/drive/folders/1BrTsbW0sy_UC4Ly7VXAiPiqCYnaUqgGK?usp=sharing