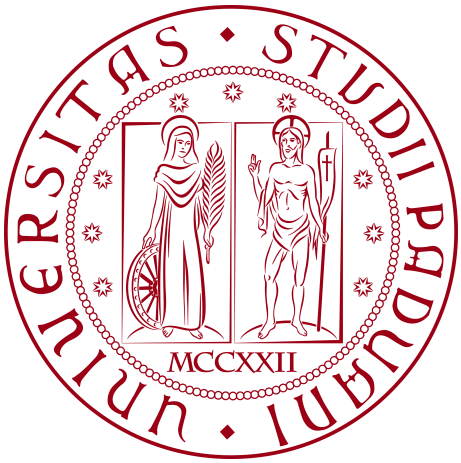
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Utilizzo dei Modelli di Machine Learning di AWS per la Classificazione e l'Estrapolazione di Informazioni contenute nelle Mail PEC

Tesi di Laurea Triennale

Relatore

Prof. Lamberto Ballan

Laureando

Riccardo Zaupa

Matricola 2034303

Anno Accademico 2023 - 2024

Ringraziamenti

Desidero esprimere la mia profonda gratitudine al professor Lamberto Ballan, relatore di questa tesi, per il prezioso supporto e la guida che mi ha offerto durante la stesura di questo lavoro. La sua disponibilità e i suoi consigli sono stati fondamentali per il raggiungimento di questo traguardo.

Un ringraziamento speciale va ai miei genitori, per il loro incondizionato sostegno, la loro pazienza e l'amore che mi hanno dimostrato durante tutto il percorso di studi. Senza il loro aiuto e la loro costante presenza, non sarei arrivato fin qui.

Infine, voglio ringraziare di cuore i miei amici per i momenti indimenticabili vissuti insieme. Gli anni trascorsi al loro fianco sono stati ricchi di avventure e di esperienze che porterò sempre con me.

Padova, Settembre 2024 Riccardo Zaupa

Sommario

Il documento presente illustra l'attività svolta durante il periodo di stage del laureando Riccardo Zaupa presso l'azienda Sanmarco Informatica S.p.A. . Tale periodo, svolto all termine del percorso di studi triennale in Informatica presso l'Università degli Studi di Padova, ha avuto una durata complessiva di trecentoventi ore.

Gli obiettivi principali del progetto si sono concentrati sull'analisi e sull'utilizzo dei servizi AWS per l'addestramento di modelli di Intelligenza Artificiale (AI), con l'intento di classificare ed estrarre automaticamente le informazioni contenute nelle mail PEC (Poste Elettroniche Certificate). Durante lo stage, è stata condotta un'analisi dettagliata dei requisiti applicativi e tecnici necessari per implementare una soluzione efficace e robusta.

L'attività di sviluppo ha incluso l'utilizzo di un modello di apprendimento automatico capace di analizzare il contenuto delle PEC importate, assegnando loro categorie appropriate e ricavandone successivamente le informazioni piu' importanti. In parallelo, è stato esplorato l'utilizzo di algoritmi avanzati di IA in grado di adattarsi e migliorare le prestazioni del modello attraverso l'apprendimento continuo dai dati e dai feedback ricevuti.

Infine, si è considerata l'integrazione con un sistema documentale per l'archiviazione automatica delle PEC, con la creazione dei metadati necessari e il loro posizionamento nella corretta categoria di appartenenza. Questi aspetti desiderabili, sebbene non obbligatori, hanno rappresentato un'opportunità di estendere la funzionalità del sistema, migliorando ulteriormente l'efficienza e l'accuratezza dell'archiviazione delle PEC.

Indice

Elenco delle figure

Elenco delle tabelle

Convenzioni tipografiche

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono evidenziati in blu e definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: [g]<termine>;
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 1

Introduzione

In questo capitolo andremo ad enunciare la struttura del documento ed analizzeremo l'azienda ospitante stage curricolare e l'offerta proposta.

1.1 Profilo aziendale

Sanmarco Informatica S.p.A. (logo in figura 1.1) è un'azienda italiana specializzata nello sviluppo software e nella consulenza informatica. Da oltre quarant'anni, Sanmarco Informatica S.p.A. si dedica alla riorganizzazione dei processi aziendali in diversi settori, progettando e implementando soluzioni digitali integrate. Con un forte orientamento verso l'innovazione, l'azienda si prefigge di agevolare la trasformazione digitale dei propri clienti, contribuendo al loro progresso.

L'innovazione rappresenta il pilastro fondamentale di Sanmarco Informatica S.p.A. . L'azienda si impegna a essere costantemente riconosciuta come altamente innovativa, investendo tra il 15% e il 20% del fatturato annuo in attività di Ricerca e Sviluppo. Questo investimento, unito alla capacità di cogliere idee e suggerimenti da clienti, dipendenti e collaboratori, alimenta lo sviluppo di nuovi prodotti e soluzioni. Particolare attenzione è riservata agli aspetti sociali e alla riduzione dell'impatto ambientale.

Oltre all'innovazione, Sanmarco Informatica S.p.A. si distingue per l'eccellenza del servizio offerto ai propri clienti. Grazie alla competenza maturata dai suoi consulenti attraverso un'esperienza pluriennale, l'azienda è in grado di proporre miglioramenti decisivi per rendere i processi aziendali più efficaci ed efficienti.

L'impegno verso la sostenibilità ambientale è un altro elemento chiave per Sanmarco Informatica S.p.A. . L'azienda ha intrapreso un percorso volto alla riduzione drastica delle proprie emissioni nocive, dimostrando un forte senso di responsabilità verso l'ambiente.

Sanmarco Informatica S.p.A. conta oggi oltre 600 dipendenti e serve più di 2500 aziende. La sede principale si trova presso Villa Romanelli a Grisignano di Zocco, in provincia di Vicenza, nelle vicinanze dei Centri di Ricerca e Sviluppo e del Centro per la Formazione di Vicenza. L'azienda dispone inoltre di filiali in Trentino-Alto Adige, Friuli-Venezia Giulia, Lombardia, Piemonte, Emilia-Romagna, Toscana, Campania e Puglia.

L'obiettivo primario di Sanmarco Informatica S.p.A. è promuovere l'innovazione e il progresso tecnologico, sviluppando soluzioni software che rispondano alle esigenze dei clienti, garantendo al contempo qualità e sicurezza. Maggiori informazioni sull'azienda sono disponibili sul sito web ufficiale[1].

Figura 1.1: Logo di Sanmarco Informatica

1.1.1 Business Unit

L'azienda è organizzata in *Business Unit* (figura 1.2), centri di competenza specifici e autonomi, ma in costante relazione tra loro. Ciascuna *Business Unit* è specializzata in un settore particolare ed è composta da team di sviluppo, consulenti e project manager, che collaborano per garantire la massima qualità e la piena soddisfazione dei clienti. Sanmarco Informatica S.p.A. dispone di 10 *Business Unit* principali:

- **Jgalileo**: La soluzione di [g]ERP che ha reso celebre Sanmarco Informatica S.p.A. . Jgalileo copre l'intero processo aziendale in modo integrato, permettendo di coordinare la filiera senza sprechi di tempo o risorse.
- **SMITech**: Specializzata nelle soluzioni per la [g]Data Protection, SMITech offre servizi di [g]Cybersecurity, gestione [g]IBM Power, progetti di infrastruttura IT, e consulenza su Privacy e [g]GDPR, migliorando la sicurezza e l'efficienza informatica delle aziende.
- **NextBI**: Questa *Business Unit* si occupa di [g]Business Intelligence, [g]Performance Management e [g]Customer Intelligence. La suite offerta da NextBI consente di gestire budget e controllo, analytics, simulazioni strategiche e di ottenere dati in tempo reale attraverso algoritmi avanzati e instant intelligence.
- **ECM**: Dedicata alla gestione della documentazione digitale, ECM (Enterprise Content Management) fornisce strumenti per gestire l'intero ciclo di vita dei contenuti elettronici.
- **Discovery Quality**: Specializzata nella gestione della [g]Governance aziendale, del sistema della Qualità e dei processi aziendali. Discovery Quality coordina in modo proattivo e strutturato i diversi enti coinvolti, siano essi interni o esterni, e permette un monitoraggio continuo grazie a potenti strumenti di analytics.
- **Factory**: Dedicata alla Supply Chain e alle Operations nella fabbrica del futuro, Factory offre una suite di software per ottimizzare la gestione della produzione, migliorare il livello di servizio ai clienti, ridurre i livelli di scorta di magazzino e massimizzare i profitti riducendo i costi. Tra gli strumenti offerti figurano il Manufacturing Execution System (JMES), l'Advanced Planning & Scheduling (APS) e il Supply Chain Collaboration (SCC).

[1] https://www.sanmarcoinformatica.com/

- **JPA**: Questa *Business Unit* è focalizzata sul Business Process Management ([g]BPM), fornendo software per creare, gestire e automatizzare i processi aziendali. JPA consente di integrare e gestire i flussi di lavoro tra diverse aree funzionali e sistemi, riducendo il margine di errore e migliorando l'efficienza operativa.
- **JPM**: JPM è il software di Project Management sviluppato per supportare le aziende nella gestione dei progetti, facilitando il raggiungimento degli obiettivi di business. Offre strumenti avanzati per la pianificazione, l'esecuzione, il monitoraggio e il controllo dei progetti, accessibili da qualsiasi dispositivo.
- **4words**: Specializzata in e-commerce, sviluppo web e app, 4words offre servizi che vanno dalla realizzazione di siti web alla creazione di applicazioni mobili, fino al posizionamento sui motori di ricerca, supportando le aziende nel loro marketing digitale.
- **TCE**: La *Business Unit* TCE è dedicata alla configurazione commerciale e tecnica, con un focus sull'ottimizzazione delle fasi di preventivazione e acquisizione degli ordini. Utilizzando la tecnologia [g]CPQ (Configure Price and Quote), TCE permette alla forza vendite di configurare offerte personalizzate, automatizzando le complesse logiche commerciali e integrandosi con strumenti di disegno tecnico e renderizzazione 3D.

La *Business Unit* di riferimento per lo stage è ECM (Enterprise Content Management), con sede presso il Centro per la Formazione di Vicenza. Questo team si occupa dello sviluppo e della manutenzione di servizi legati alla gestione dei documenti digitali, come ad esempio: la gestione documentale, Discovery XChange (per la conformità normativa sulla fatturazione elettronica), conservazione [g]PEC (Posta Elettronica Certificata) integrata con Aruba, Digifinder (strumento per la ricerca di fatture elettroniche), Firmae (servizio di firma digitale dei documenti), e molti altri.

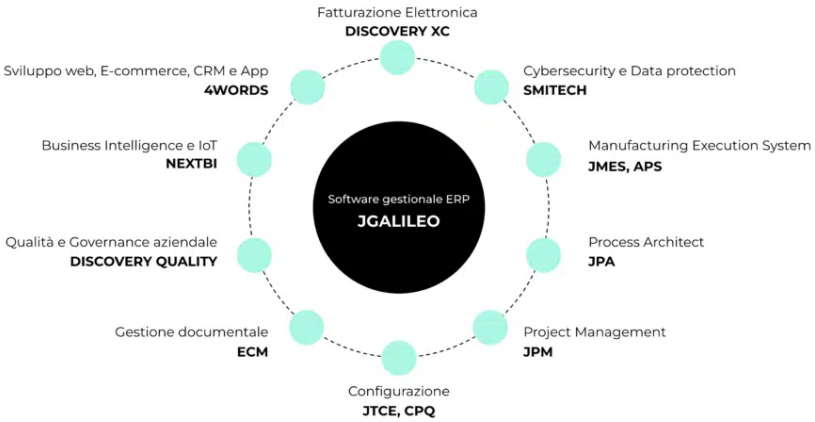


Figura 1.2: Le Business Unit di Sanmarco Informatica

1.1.2 Metodologia di sviluppo

La metodologia di lavoro, indipendentemente dalla Business Unit, è basata su un approccio [g]Agile implementata con il framework [g]Scrum. Agile è un approccio alla gestione dei progetti che si fonda su principi di collaborazione, auto-organizzazione e flessibilità. Scrum (figura 1.3), in particolare, è un framework Agile che facilita la gestione di progetti complessi, garantendo la massima trasparenza e flessibilità. Questo viene realizzato suddividendo il progetto in sprint, ovvero periodi di tempo relativamente brevi in cui vengono fissati determinati obiettivi e attività.

Scrum si basa sull'idea che le decisioni si prendano in base a ciò che è noto al momento. Per questo motivo, la metodologia prevede tre principi fondamentali:

- **Trasparenza**: Il team deve utilizzare un linguaggio comune e ben definito, per evitare dubbi e incomprensioni.
- **Ispezione**: Gli avanzamenti del progetto vengono ispezionati frequentemente per assicurarsi che il prodotto rimanga conforme ai requisiti e non si discosti dagli obiettivi.
- **Adattamento**: Se vengono rilevate discrepanze, il team deve essere in grado di adattarsi rapidamente, modificando le parti non conformi del prodotto per ridurre al minimo lo scarto rispetto agli obiettivi stabiliti. Questo processo avviene durante momenti specifici come lo Sprint Planning Meeting, il Daily Scrum e lo Sprint Review.

Il ciclo di sviluppo Scrum è strutturato in sprint, che sono iterazioni brevi, solitamente di durata variabile tra una e quattro settimane. Ogni sprint inizia con un meeting di pianificazione (*Sprint Planning Meeting*), durante il quale il team di sviluppo esamina il lavoro svolto nello sprint precedente e definisce i nuovi obiettivi. Gli obiettivi fissati per uno sprint non possono essere modificati durante la sua esecuzione. Al termine dello sprint, il team presenta al committente una versione funzionante del prodotto, che include i progressi realizzati.

Per ogni sprint, i requisiti vengono estratti dal *Product Backlog* e inseriti nello *Sprint Backlog*, dove vengono suddivisi in task (o ticket), ciascuno dei quali rappresenta un'unità di lavoro da completare in un giorno.

Per garantire la fattibilità del prodotto e mantenere un'organizzazione ottimale, Scrum prevede una serie di eventi regolari:

- **Sprint Planning Meeting**: All'inizio di ogni sprint, il team di sviluppo si riunisce per selezionare il lavoro da svolgere e definire il tempo necessario per ogni requisito del *Product Backlog*.
- **Daily Scrum**: Durante lo sprint, il team si incontra quotidianamente per un breve meeting di massimo 15 minuti, in cui si discute del lavoro svolto, di quello previsto per la giornata successiva e degli eventuali problemi riscontrati.
- **Sprint Review**: Alla fine dello sprint, il team si riunisce con il committente per esaminare i cambiamenti nei requisiti, valutare il lavoro svolto e discutere delle problematiche riscontrate.

Questo incontro è fondamentale per generare nuove idee e migliorare il prodotto, contribuendo alla pianificazione dello sprint successivo.

Ogni team Scrum è strutturato per essere indipendente, con l'obiettivo di ottimizzare i feedback ricevuti dal committente. Il team è generalmente composto da tre ruoli principali:

- **Product Owner**: La persona che commissiona il progetto, segue lo sviluppo del prodotto e ne definisce i requisiti.
- **Team di sviluppo**: Il gruppo di professionisti che lavora alla realizzazione del prodotto.
- **Scrum Master**: La figura responsabile di mantenere il team di sviluppo focalizzato e motivato, proteggendolo da distrazioni e ponendo sfide che favoriscano il miglioramento continuo. Questo ruolo è spesso distinto dal Project Manager.

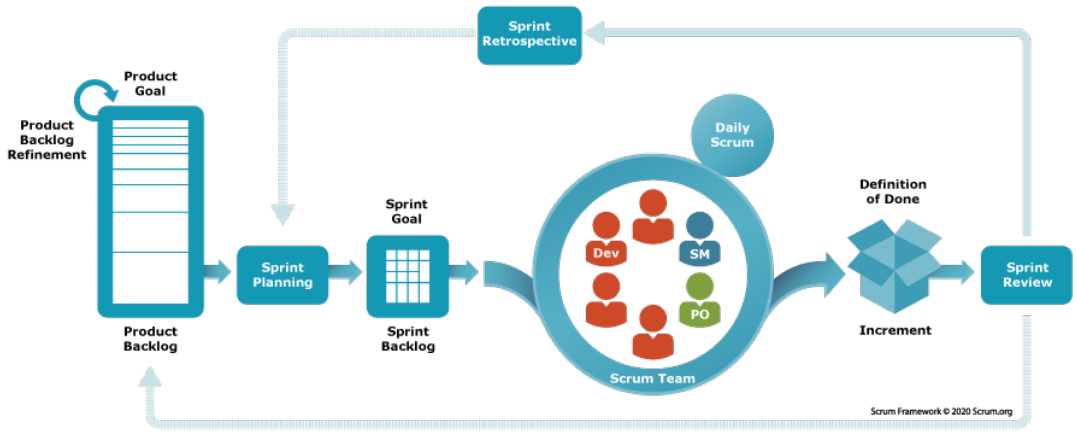


Figura 1.3: Framework Scrum

1.2 L'offerta di stage

L'obiettivo dello stage consiste nello sviluppo di un sistema avanzato per la catalogazione delle Poste Elettroniche Certificate (PEC), integrando tecnologie di [g]Artificial Intelligence (AI) per migliorare l'efficienza e l'accuratezza del processo.

Gli obiettivi del progetto in fase di proposta dello stage sono i seguenti:

- **Catalogazione Automatica Potenziata dall'IA**: Implementare modelli di apprendimento automatico che analizzino il contenuto delle PEC importate e le classifichino automaticamente in base al contenuto. L'AI sarà in grado di rilevare informazioni quali mittente, destinatario, data e argomento, migliorando significativamente l'efficienza del processo di catalogazione.

- **Integrazione con un Sistema di Gestione Documentale**: Le informazioni estratte dalle PEC saranno integrate con un sistema di gestione documentale (DocuWave). Questo permetterà di persistere le email nel sistema, creando i metadati necessari con le informazioni estratte e collocandole nella corretta categoria di appartenenza.
- **Adattamento e Apprendimento Continuo**: Saranno implementati algoritmi di AI in grado di adattarsi e apprendere continuamente dai dati, migliorando le prestazioni del sistema nel tempo. Questo processo includerà l'ottimizzazione dei modelli di apprendimento automatico in base all'esperienza acquisita e ai feedback degli utenti.
- **Utilizzo dei Servizi Cloud AWS**: Il progetto prevede l'utilizzo dei servizi cloud offerti da [g]Amazon Web Services (AWS) per l'addestramento del modello di apprendimento automatico scelto e per l'erogazione del servizio. L'infrastruttura necessaria sarà configurata direttamente sul cloud AWS per garantire scalabilità ed efficienza.

Il progetto è stato proposto dall'azienda in occasione dell'evento Stage IT 2024, organizzato dall'Università degli Studi di Padova e promosso da Confindustria Veneto Est. Questo evento mira a facilitare l'incontro tra studenti e aziende, offrendo la possibilità di svolgere uno stage formativo con particolare riferimento al settore [g]ICT.

1.3 Organizzazione del testo

Il secondo capitolo descrive lo stage, l'organizzazione, gli obiettivi e le attività svolte.

Il terzo capitolo approfondisce le tecnologie utilizzate oltre che gli strumenti e le metodologie di lavoro.

Il quarto capitolo approfondisce la progettazione e la codifica del progetto.

Il quinto capitolo descrive i possibili sviluppi futuri del progetto.

Il sesto capitolo descrive le conclusioni del lavoro svolto.

Capitolo 2

Descrizione dello stage

In questo capitolo viene presentata una panoramica del progetto di stage, con una descrizione dettagliata del contesto aziendale, del progetto e degli obiettivi prefissati. Vengono inoltre elencati i requisiti e gli obiettivi del progetto, i prodotti attesi e la pianificazione delle attività.

2.1 Introduzione al progetto

Le organizzazioni di diversi settori, come sanità, finanza, legale, retail e manifatturiero, gestiscono quotidianamente una grande quantità di documenti nei loro processi aziendali. Questi documenti contengono informazioni critiche, essenziali per prendere decisioni tempestive e mantenere alti livelli di soddisfazione del cliente, velocizzare l'onboarding e ridurre il tasso di abbandono dei clienti. Nella maggior parte dei casi, l'elaborazione di tali documenti avviene manualmente, un processo che richiede tempo, è soggetto a errori, costoso e difficile da scalare. Inoltre, l'automazione attualmente disponibile per l'elaborazione dei documenti è limitata.

L'elaborazione intelligente dei documenti ([g]Intelligence Document Processing (IDP)) consente di automatizzare l'estrazione delle informazioni da documenti di diverso tipo e formato, in modo rapido e preciso, senza necessità di competenze avanzate di [g]Machine Learning (ML). Questa tecnologia riduce i costi complessivi, migliorando al contempo l'efficienza e la qualità delle decisioni aziendali.

IDP automatizza la raccolta e l'elaborazione delle informazioni dai documenti digitali o cartacei, integrandole nei flussi di lavoro aziendali digitali. Ad esempio, in un'azienda che invia ordini ai fornitori al calo delle scorte, IDP sostituisce l'immissione manuale dei dati estraendo automaticamente le informazioni rilevanti dalle fatture ricevute via e-mail e integrandole nel sistema contabile. Questo processo elimina gli ostacoli e riduce notevolmente gli errori, migliorando l'efficienza operativa.

IDP offre numerosi vantaggi alle aziende, come la [g]Scalabilità nella gestione di grandi volumi di documenti, l'automazione delle attività ripetitive e la riduzione dei costi di elaborazione manuale. Inoltre, velocizza l'interazione con i clienti, automatizzando attività come l'onboarding e la gestione dei pagamenti, garantendo risposte tempestive e personalizzate.

I settori che beneficiano dell'IDP includono la sanità, dove facilita la gestione delle cartelle cliniche e l'organizzazione dei dati medici; le finanze, dove automatizza la gestione delle spese e delle fatture; il settore legale, dove analizza contratti e documenti complessi; la logistica, dove migliora la tracciabilità delle spedizioni; e le risorse umane, dove semplifica la selezione del personale e gestisce le buste paga.

Le tecnologie principali che supportano l'IDP includono il riconoscimento ottico dei caratteri ([g]Optical Character Recognition (OCR)), che converte le immagini di testo in dati leggibili dalle macchine, e l'elaborazione del linguaggio naturale ([g]Natural Language Processing (NLP)), che analizza e comprende il linguaggio umano. L'automazione robotica dei processi ([g]Robotic Process Automation (RPA)) permette di ripetere azioni umane predefinite per automatizzare i flussi di lavoro aziendali.

Il processo di IDP (vedi Figura 2.1) segue tipicamente diverse fasi: acquisizione e classificazione dei documenti, estrazione dei dati tramite OCR e NLP, convalida e inserimento delle informazioni nei sistemi aziendali, e apprendimento continuo attraverso modelli di ML per migliorare l'accuratezza. I sistemi di IDP forniscono anche report e analisi che aiutano le aziende a ottimizzare ulteriormente i loro flussi di lavoro.

In un contesto aziendale sempre più orientato all'innovazione, l'elaborazione dei documenti ha subito notevoli trasformazioni grazie all'introduzione dell'IDP, che trasforma i dati non strutturati presenti in vari tipi di documenti in informazioni strutturate e fruibili, migliorando drasticamente l'efficienza e riducendo lo sforzo manuale. Tuttavia, il potenziale dell'IDP non si esaurisce qui. Con l'integrazione dell'intelligenza artificiale generativa ([g]Generative AI), l'IDP può essere ulteriormente potenziata.

L'integrazione di modelli di linguaggio di grandi dimensioni ([g]LLM) nelle architetture IDP consente di ottenere capacità avanzate di estrazione dati, adattandosi dinamicamente ai cambiamenti nei modelli dei dati. Amazon Web Services AWS supporta questa evoluzione con strumenti come Amazon Textract, un servizio di machine learning (ML) che estrae automaticamente testo, scrittura e dati da documenti scansionati, e Amazon Bedrock, un servizio completamente gestito che offre modelli di intelligenza artificiale di base attraverso [g]API di facile utilizzo. Un ulteriore strumento chiave è Amazon Comprehend, un servizio di NLP che consente di analizzare e comprendere il contenuto dei documenti, estraendo informazioni chiave, come entità, sentiment e concetti, direttamente dal testo.

In particolare, l'integrazione di Amazon Textract con LangChain, utilizzato come document loader, e Amazon Bedrock, per l'estrazione di dati e capacità di AI, permette di estendere le funzionalità di una nuova o esistente architettura IDP. Amazon Comprehend si combina perfettamente in questo flusso, utilizzando NLP per analizzare il contenuto dei documenti e fornire insight dettagliati, aumentando la precisione dell'estrazione delle informazioni e permettendo di ottenere analisi più approfondite.

Questa combinazione introduce non solo un'automazione più avanzata nell'elaborazione dei documenti, ma anche una capacità di adattamento e miglioramento continuo grazie all'AI, consentendo di affrontare in modo dinamico i modelli di dati in evoluzione.

In conclusione, l'integrazione di IDP, AI e strumenti come Amazon Textract, Amazon Comprehend e Amazon Bedrock rappresenta una nuova frontiera nell'elaborazione documentale, offrendo alle aziende non solo efficienza operativa, ma anche flessibilità e capacità di risposta alle nuove sfide del mercato.

Nel contesto del progetto di stage, l'obiettivo è implementare un modello di AI per l'elaborazione

intelligente degli allegati delle PEC e per perseguire a questo scopo, si è deciso di utilizzare i servizi AWS per l'addestramento del modello di AI e per l'elaborazione delle PEC importate.

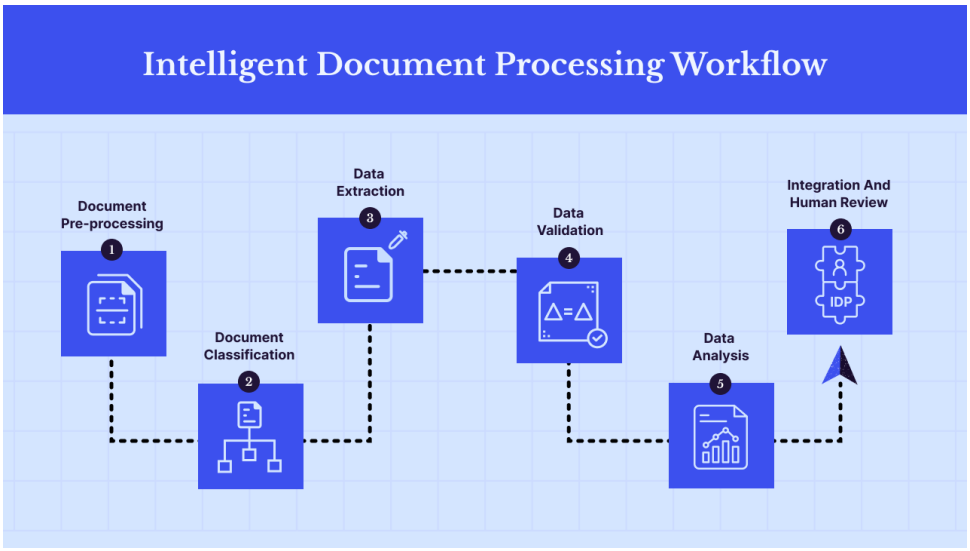


Figura 2.1: Flusso di lavoro dell'elaborazione intelligente dei documenti

2.2 Requisiti e obiettivi

Gli obiettivi del progetto sono stati definiti in accordo con il tutor aziendale e si articolano nel modo seguente:

[Priorità][Id]

- **Priorità**: indica il livello di importanza dell'obiettivo, che può essere *Obbligatorio* o *Desiderabile*;
- **Id**: composto da due cifre, identifica in modo univoco l'obiettivo rispetto alla priorità.

Di seguito è riportata una tabella che elenca i requisiti e gli obiettivi stabiliti per il progetto.

ID	Categoria	Descrizione
O01	Obbligatorio	Analisi dei servizi AWS per l'addestramento dei modelli di AI.
O02	Obbligatorio	Addestramento di un modello di apprendimento AI utilizzando i servizi AWS.
O03	Obbligatorio	Analisi dei requisiti applicativi e tecnici per implementare la soluzione richiesta.
O04	Obbligatorio	Implementazione di un modello di apprendimento automatico che analizzi il contenuto delle PEC importate e assegni loro categorie appropriate (mittente, destinatario, data e argomento).

ID	Categoria	Descrizione
D01	Desiderabile	Implementazione di algoritmi di AI in grado di adattarsi e apprendere continuamente dai dati per migliorare le prestazioni del sistema nel tempo. Questo include l'ottimizzazione dei modelli di apprendimento in base all'esperienza e ai feedback degli utenti.
D02	Desiderabile	Integrazione con un sistema documentale per l'archiviazione delle PEC, creando i metadati necessari e assegnando le informazioni estratte alla corretta categoria.

Tabella 2.1: Tabella dei requisiti e obiettivi dello stage

2.2.1 Prodotti attesi

Tra i principali risultati attesi dal progetto, lo studente dovrà produrre una relazione scritta che illustri in dettaglio i punti chiave del lavoro svolto. In particolare, la relazione dovrà includere:

- Una contestualizzazione del progetto, che spieghi il problema affrontato e gli obiettivi perseguiti;
- Un'analisi completa e approfondita che copra:
  - L'inquadramento generale del progetto;
  - I requisiti applicativi e tecnici;
  - La struttura del database;
  - Gli strumenti e le applicazioni di terze parti utilizzati;
  - I principali casi d'uso individuati.
- Uno studio di fattibilità, volto a dimostrare la possibilità di implementare la soluzione proposta;
- La descrizione dettagliata dell'implementazione della soluzione sviluppata.

2.2.2 Contenuti formativi previsti

Nel corso di questo progetto di stage, lo studente avrà l'opportunità di approfondire diverse aree tecniche e migliorare le proprie competenze. In particolare, gli ambiti di conoscenza che saranno oggetto di apprendimento includono:

- **Linguaggi e strumenti tecnologici**: lo studente acquisirà familiarità con una serie di tecnologie chiave, tra cui:
  - L'uso del database MySQL, se necessario per il progetto;
  - I framework più diffusi come Angular, Maven, Spring e AWS [g]SDK;
  - Diversi linguaggi di programmazione e formati di dati, quali Java, JSON, HTML, CSS, Javascript, Typescript e il formato PDF/A2;

  - I servizi cloud di AWS, utilizzati per l'addestramento dei modelli di AI;
  - Python e gli strumenti di analisi forniti da notebook Jupyter.

- **Competenze trasversali**: lo studente acquisirà inoltre esperienza pratica nel lavoro di gruppo, in particolare all'interno di un team che utilizza il framework Agile Scrum, migliorando così le proprie capacità di collaborazione e gestione del lavoro in un contesto aziendale strutturato.

2.3 Pianificazione

La pianificazione delle attività è stata organizzata in base agli obiettivi prefissati e alle scadenze stabilite. La tabella seguente riporta la pianificazione settimanale delle attività svolte durante il periodo di stage.

Settimana	Dal	Al	Attività
1	24-06-2024	28-06-2024	Incontro con le persone coinvolte nel progetto per discutere i requisiti e le richieste di implementazione; ricerca, studio e documentazione per l'inquadramento del progetto; introduzione ai linguaggi di sviluppo; introduzione agli ambienti di sviluppo; introduzione dei servizi AWS.
2	01-07-2024	05-07-2024	Analisi dei servizi AWS per l'addestramento di un modello di apprendimento; addestramento di un modello di apprendimento utilizzando i servizi di AWS. **Milestone:** Utilizzo dei servizi AWS per l'addestramento di un modello di apprendimento.
3	08-07-2024	12-07-2024	Studio della soluzione per definire i requisiti necessari per l'implementazione. **Milestone:** Analisi dei requisiti applicativi e tecnici per implementare la soluzione.
4	15-07-2024	19-07-2024	Addestramento del modello di apprendimento per catalogare le PEC in base al loro contenuto.
5	22-07-2024	26-07-2024	Implementazione per interfacciarsi con il modello di apprendimento addestrato e per catalogare le PEC importate. **Milestone:** Completamento degli obiettivi minimi.
6	29-07-2024	02-08-2024	Implementazione dell'algoritmo di AI per l'autoapprendimento.

Settimana	Dal	Al	Attività
7	05-08-2024	09-08-2024	Studio e documentazione sulle [g]API messe a disposizione dal sistema documentale per catalogare le mail PEC; implementazione dell'integrazione con il documentale producendo i metadati necessari per catalogare le PEC.
8	12-08-2024	16-08-2024	Verifica e test dell'archiviazione delle PEC nel documentale. **Milestone:** Completamento degli obiettivi massimi.
9	19-08-2024	23-08-2024	Recupero di eventuali ritardi.
10	26-08-2024	30-08-2024	Recupero di eventuali ritardi.

Tabella 2.2: Tabella della pianificazione dello stage

Durante lo svolgimento dello stage, sono state applicate modifiche alla pianificazione in base alle necessità e alle richieste del tutor aziendale.

2.4 Organizzazione del lavoro

Lo stage si è svolto nel periodo dal 24 giugno 2024 al 30 agosto 2024, con una durata complessiva di 8 settimane e un totale di 320 ore di lavoro. È stata prevista una modalità mista, con 2 giorni a settimana di presenza in azienda e 3 giorni in modalità remota, con impegno full-time ogni giorno. La sede di riferimento è stata quella di Sanmarco Informatica S.p.A. situata in Via dell'Edilizia, 100, Vicenza (VI), ovvero il Centro per la Formazione.

Lo studente è stato inserito in un gruppo di lavoro, con il supporto continuo del team e del tutor aziendale. Il tutor è stato spesso presente nel gruppo, garantendo una modalità di interazione costante e facilitando il processo di revisione e feedback. Le revisioni del progetto sono state condotte in conformità alla metodologia Scrum, con brevi riunioni giornaliere di 5 minuti e una revisione settimanale della durata di 1 ora.

L'azienda ha fornito strumenti di comunicazione e collaborazione come *Google Meet* per le riunioni e *Google Drive* per la condivisione dei documenti, organizzati attraverso un gruppo creato su Gmail. La comunicazione in modalità smart working è avvenuta tramite la chat di *Google Chat*, garantendo un canale diretto per il dialogo tra lo studente e il tutor aziendale.

Oltre ai punti sopra elencati, durante il corso dello stage sono stati svolti due incontri con il tutor aziendale, denominati SAL (Stato Avanzamento Lavoro), per discutere lo stato di avanzamento del progetto e valutare eventuali modifiche da apportare. Il primo incontro è stato svolto a metà stage, mentre il secondo è avvenuto alla conclusione dello stage. L'incontro di chiusura è stato utilizzato per discutere i risultati ottenuti e per valutare il lavoro complessivo svolto.

Capitolo 3

Tecnologie e strumenti di interesse

In questo capitolo verranno descritti i servizi e le tecnologie analizzate e pertinenti per il problema descritto, in quale modo possono essere impiegate e una panoramica finalizzata a chiarirne il contesto e il caso d'uso.

3.1 Amazon Web Services

AWS è una piattaforma di servizi cloud che offre potenza di calcolo, storage di database, distribuzione di contenuti e altre funzionalità per aiutare le aziende a scalare e crescere. AWS offre una vasta gamma di servizi che possono essere utilizzati per implementare soluzioni di AI e ML e in particolare che possano implementare un flusso di IDP automatizzato e adatto agli obiettivi del progetto.
Per la realizzazione dell'applicazione sono stati individuati diversi servizi che hanno permesso di realizzare un'architettura scalabile e serverless.

3.1.1 Amazon Comprehend

Amazon Comprehend (il logo è riportato in Figura 3.1) è un servizio avanzato di analisi del linguaggio naturale (NLP) che utilizza algoritmi di apprendimento automatico per estrarre informazioni significative dai testi. Il servizio è in grado di identificare entità, frasi chiave, lingua, sentimenti e altre caratteristiche comuni all'interno dei documenti, offrendo la possibilità di effettuare analisi sia in tempo reale che in modalità asincrona su grandi volumi di dati. Gli utenti possono scegliere di utilizzare modelli pre-addestrati o di addestrare modelli personalizzati per specifiche esigenze di classificazione e riconoscimento delle entità.
Tra le principali funzionalità di Amazon Comprehend vi è *Amazon Comprehend Insights*, che consente di analizzare documenti, singoli o in gruppo, per identificare le informazioni più rilevanti utilizzando modelli già addestrati. Questi modelli possono essere impiegati per individuare entità (come persone, luoghi, date, quantità, ecc.), frasi chiave, informazioni personali identificabili, sentiment (positivo, negativo, neutro) oltre a determinare la lingua e la sintassi del testo.
Un'altra funzionalità rilevante è *Amazon Comprehend Custom*, che permette la creazione di modelli NLP

personalizzati per la classificazione (*Custom Classification*) e il riconoscimento delle entità (*Custom Entity Recognition*). La *Custom Classification* consente di categorizzare i documenti in base a categorie predefinite, mentre la *Custom Entity Recognition* permette di individuare entità specifiche all'interno dei testi. Entrambi i servizi richiedono una fase di training che necessita di un [g]dataset etichettato per addestrare il modello e supportano l'elaborazione dei documenti in un'unica fase.

In aggiunta, Amazon Comprehend offre la funzionalità *Flywheel*, che semplifica il processo di addestramento e gestione delle versioni dei modelli personalizzati, facilitando l'orchestrazione delle attività di training, valutazione e deployment dei modelli. Consiste dunque nel riferimento principale per la fase di [g]Machine Learning Operations (MLOps) e permette di monitorare le prestazioni dei modelli, valutare le metriche di accuratezza e precisione e gestire le versioni dei modelli in produzione.

Infine, il *Document Clustering* permette di raggruppare i documenti in base a parole chiave ricorrenti, rendendo più agevole l'identificazione di documenti simili e la loro organizzazione per categorie o argomenti.

Nel presente lavoro, Amazon Comprehend è stato utilizzato per la classificazione dei documenti nelle categorie selezionate tramite la funzionalità *Custom Classification*.

Figura 3.1: Logo di Amazon Comprehend

3.1.2 Amazon Textract

Amazon Textract (il logo è riportato in Figura 3.2) è un servizio di riconoscimento ottico dei caratteri (OCR) che sfrutta l'apprendimento automatico per identificare e analizzare testo e dati presenti in immagini o documenti. Basato sulla tecnologia di [g]Deep Learning collaudata e altamente scalabile sviluppata dagli esperti di [g]Computer Vision di Amazon, Textract è in grado di analizzare quotidianamente miliardi di immagini e video. Una delle caratteristiche distintive di questo servizio è la sua accessibilità: non è richiesta alcuna esperienza nel campo del ML per utilizzarlo, grazie alla disponibilità di API semplici e intuitive che consentono di analizzare file immagine e PDF con facilità. Inoltre, Amazon Textract apprende continuamente dai nuovi dati e Amazon implementa costantemente nuove funzionalità, garantendo un miglioramento continuo delle sue capacità.

Il servizio non si limita a eseguire il riconoscimento ottico dei caratteri da testo digitato o scritto a mano, ma è anche in grado di estrarre il contenuto del documento, incluse tabelle, campi e relazioni strutturali. Textract fornisce punteggi di confidenza e bounding box (rappresentazioni grafiche dei confini) per ogni parola e riga di testo riconosciuta. Il servizio supporta vari formati di file, tra cui PDF, TXT, DOC, DOCX, JPG e PNG.

Le principali funzionalità di Amazon Textract includono:

- **Estrazione di testo non strutturato**: Questa funzionalità consente di estrarre i dati in forma di parole (*WORDS*) e righe di testo (*LINES*), senza mantenere la formattazione originaria del documento. Per questa operazione si utilizza l'API `DetectDocumentText`.
- **Estrazione ed elaborazione di moduli e tabelle**: Tramite l'API `AnalyzeDocument`, è possibile estrarre dati mantenendo la struttura del documento originale, identificando parole, righe, tabelle e moduli (*WORDS*, *LINES*, *TABLES*, *FORMS*).
- **Estrazione di coppie chiave-valore**: Utilizzando l'API `AnalyzeDocument`, questa funzionalità permette di estrarre informazioni strutturate in forma di chiavi e valori, preservando la formattazione del documento.
- **Estrazione tramite query**: Questa funzionalità consente di focalizzarsi su informazioni specifiche o critiche all'interno di un documento. Anche in questo caso, l'API utilizzata è `AnalyzeDocument`.
- **Rilevamento delle firme**: Attraverso l'API `AnalyzeDocument`, è possibile rilevare la presenza di firme nei documenti, restituendo un punteggio di confidenza per il rilevamento, oltre al testo del documento in forma di parole e righe (*WORDS* e *LINES*).
- **Estrazione di informazioni da fatture e ricevute**: L'API `AnalyzeExpense` è specificamente progettata per estrarre dati da documenti contabili come fatture e ricevute.
- **Estrazione di informazioni da documenti di identità**: Utilizzando l'API `AnalyzeID`, è possibile estrarre dati rilevanti da documenti di identità.
- **Rilevamento di testo su più colonne**: Questa funzionalità consente di riconoscere e trattare testi distribuiti su più colonne all'interno di un documento.

Per migliorare la precisione delle analisi e ridurre l'intervento umano necessario, Amazon Textract offre lo strumento delle *Custom Queries*. Questo strumento consente di riconoscere specifici termini univoci, strutture particolari e informazioni specifiche all'interno dei documenti, offrendo un livello di personalizzazione superiore rispetto alle query standard.

Un'altra opzione avanzata per personalizzare l'output dell'analisi dei documenti è l'uso degli *Adapters*. Gli Adapters sono componenti che si integrano nel modello di Deep Learning pre-addestrato di Amazon Textract, permettendo di personalizzare l'output in base ai documenti specifici di un'azienda. Per creare un Adapter, è necessario annotare ed etichettare un insieme di documenti campione e addestrare l'Adapter su questi campioni annotati.

Una volta creato un Adapter, Amazon Textract fornisce un *AdapterId*. È possibile creare e gestire diverse versioni di un Adapter all'interno di uno stesso identificatore. L'*AdapterId*, insieme alla versione dell'Adapter, può essere utilizzato in una richiesta per specificare l'uso dell'Adapter creato durante l'analisi dei documenti. Ad esempio, questi parametri possono essere forniti all'API `AnalyzeDocument`

per un'analisi sincrona dei documenti, oppure all'operazione `StartDocumentAnalysis` per un'analisi asincrona. Includendo l'*AdapterId* nella richiesta, l'Adapter verrà automaticamente integrato nel processo di analisi, migliorando le previsioni per i documenti specifici.

Questo approccio consente di sfruttare le capacità dell'API `AnalyzeDocument` mentre si adatta il modello alle esigenze specifiche del proprio caso d'uso.

Nel contesto del presente lavoro, Amazon Textract è stato utilizzato per estrarre il testo dai documenti sia come input al classificatore di Comprehend sia per estrarre informazioni utili.

Figura 3.2: Logo di Amazon Textract

3.1.3 Amazon S3

Amazon Simple Storage Service (Amazon S3) (logo riportato in Figura 3.3) è un servizio di storage di oggetti che offre elevata scalabilità, disponibilità dei dati, sicurezza e prestazioni. Amazon S3 è progettato per gestire grandi volumi di dati a costi contenuti, risultando una soluzione ideale per applicazioni che richiedono capacità di archiviazione massiva.

Per memorizzare dati in Amazon S3, è necessario utilizzare un *bucket*, che funge da contenitore per gli oggetti. Ogni oggetto in un *bucket* rappresenta un file e i relativi metadati associati. La procedura per archiviare un oggetto in Amazon S3 prevede la creazione di un *bucket* e il successivo caricamento dell'oggetto al suo interno. Una volta caricato, l'oggetto può essere aperto, scaricato o eliminato. Qualora un oggetto o un *bucket* non siano più necessari, è possibile procedere alla loro eliminazione.

Nel contesto del presente progetto, Amazon S3 è stato utilizzato per memorizzare i file relativi alle diverse fasi del lavoro, inclusi allegati, email, file CSV impiegati per l'addestramento dei modelli e file di output generati dalle analisi.

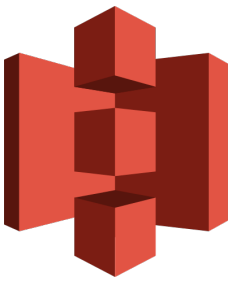


Figura 3.3: Logo di Amazon S3

3.1.4 AWS Lambda

AWS Lambda (logo riportato in Figura 3.4) è un servizio di calcolo serverless che esegue codice in risposta a eventi, gestendo automaticamente le risorse di calcolo necessarie. Questo servizio elimina la necessità di provisioning e gestione dei server, offrendo una soluzione scalabile e affidabile per diverse applicazioni.

Il codice in Lambda è organizzato in funzioni che vengono eseguite solo quando richiesto, scalando automaticamente in base al carico. La tariffazione si basa esclusivamente sul tempo di calcolo utilizzato, senza costi aggiuntivi quando il codice non è in esecuzione. Questa flessibilità lo rende ideale per scenari che richiedono scalabilità dinamica e riduzione automatica delle risorse in assenza di carico.

Nel contesto del presente progetto, AWS Lambda è stato impiegato per implementare le funzioni di chiamate API, garantendo un'architettura serverless efficiente. Le funzioni Lambda sono state integrate con altri servizi AWS, come Amazon S3 per l'elaborazione dei file e Amazon API Gateway per la gestione delle richieste API. L'adozione di Lambda ha permesso di semplificare la gestione operativa, poiché il servizio si occupa automaticamente di capacità, monitoraggio e logging, lasciando agli sviluppatori la responsabilità esclusiva del codice.

Figura 3.4: Logo di AWS Lambda

3.1.5 Amazon DynamoDB

Amazon DynamoDB (logo riportato in Figura 3.5) è un servizio di database [g]NoSQL completamente gestito, progettato per garantire prestazioni a singola cifra di millisecondi indipendentemente dalla scala. Ideale per carichi di lavoro operativi che richiedono alta efficienza, DynamoDB affronta le complessità di scalabilità e gestione operativa tipiche dei database relazionali, mantenendo prestazioni elevate anche in presenza di un grande numero di utenti. Questo lo rende particolarmente adatto per applicazioni moderne che necessitano di crescere rapidamente a livello globale.

Dal suo lancio nel 2012, DynamoDB è stato adottato da organizzazioni di ogni settore e dimensione per sviluppare applicazioni che possono iniziare con piccoli volumi di dati e scalare fino a supportare tabelle di dimensioni virtualmente illimitate, assicurando al contempo alta disponibilità.

Nel contesto del presente progetto, Amazon DynamoDB è stato utilizzato per la memorizzazione dei dati estratti dai documenti e delle classificazioni effettuate, garantendo un accesso rapido e affidabile alle informazioni archiviate.

Figura 3.5: Logo di Amazon DynamoDB

3.1.6 AWS Step Functions

AWS Step Functions (logo riportato in Figura 3.6) è un servizio di orchestrazione serverless che consente di coordinare in modo efficiente i componenti di applicazioni distribuite, microservizi e pipeline di dati o di ML attraverso una logica visuale. Questo servizio si basa sul concetto di macchine a stati (*State machines*) e task, dove una macchina a stati, o workflow, è costituita da una serie di passaggi guidati da eventi. Ogni passaggio nel workflow è chiamato stato, e uno stato di tipo Task rappresenta un'unità di lavoro eseguita da un altro servizio AWS o API. Le esecuzioni, ovvero le istanze di workflow in esecuzione, sono gestite direttamente da Step Functions.

Le attività all'interno dei task della macchina a stati possono anche essere svolte utilizzando le *Activities*, che sono lavoratori esterni al servizio Step Functions.

Nel contesto del presente progetto, AWS Step Functions è stato utilizzato per orchestrare i vari servizi AWS coinvolti, in particolare le funzioni Lambda.

Figura 3.6: Logo di Amazon Step Functions

3.1.7 Amazon SageMaker

Amazon SageMaker (logo riportato in Figura 3.7) è un servizio completamente gestito per il ML che permette a data scientist e sviluppatori di costruire, addestrare e distribuire modelli ML in un ambiente di produzione altamente scalabile e sicuro. SageMaker facilita l'intero processo di sviluppo di modelli ML, fornendo un'interfaccia utente intuitiva che integra strumenti e funzionalità di ML all'interno di diversi ambienti di sviluppo integrato ([g]IDE).

SageMaker consente di archiviare e condividere i dati senza dover gestire infrastrutture server, permettendo alle organizzazioni di concentrarsi sullo sviluppo collaborativo dei flussi di lavoro ML. Il servizio supporta algoritmi ML gestiti, ottimizzati per elaborare grandi volumi di dati in un ambiente distribuito, e offre la flessibilità di utilizzare algoritmi e framework personalizzati. In pochi passaggi, è possibile distribuire un modello in un ambiente sicuro e scalabile direttamente dalla console di SageMaker.

Tra gli strumenti offerti da Amazon SageMaker vi sono:

- **Amazon SageMaker JumpStart**: Un hub di ML che consente di valutare e selezionare modelli fondamentali (*foundation models*) in base a specifici parametri.
- **Amazon SageMaker Studio**: Un IDE completo per preparare i dati, creare, addestrare e distribuire modelli ML, offrendo strumenti per ogni fase del ciclo di vita del ML.
- **Amazon SageMaker MLOps**: Fornisce strumenti per automatizzare le operazioni di ML lungo tutto il ciclo di vita del modello, inclusi processi di integrazione e distribuzione continua (CI/CD).
- **Amazon SageMaker BlazingText**: Implementa l'algoritmo Word2Vec per la creazione di vettori di parole, utilizzati nell'elaborazione del linguaggio naturale.
- **Pipeline di Amazon SageMaker**: Automatizza le diverse fasi del ML, dalla pre-elaborazione dei dati al monitoraggio dei modelli in produzione.
- **Amazon SageMaker Ground Truth**: Migliora la precisione dei modelli ML sfruttando il feedback umano durante tutto il ciclo di vita del modello, permettendo anche la creazione di etichette per i dati.
- **Amazon SageMaker Clarify**: Rileva e mitiga i pregiudizi presenti nei dati di addestramento e nelle previsioni dei modelli ML.
- **Amazon SageMaker Model Monitor**: Monitora i modelli ML in produzione per rilevare eventuali cambiamenti nei dati o nelle prestazioni dei modelli, assicurando un'accuratezza costante nel tempo.

Nel contesto del presente progetto, Amazon SageMaker non è stato utilizzato direttamente, in quanto si è ritenuto l'utilizzo di Amazon Comprehend e Amazon Textract sufficiente per le esigenze di analisi del testo e dei documenti. Tuttavia, SageMaker rappresenta una risorsa fondamentale per lo sviluppo di modelli ML personalizzati e per l'implementazione di soluzioni di ML avanzate.

Figura 3.7: Logo di Amazon SageMaker

3.1.8 Amazon Bedrock

Amazon Bedrock (logo riportato in Figura 3.8) è un servizio completamente gestito che offre una selezione di modelli di fondazione [g]FM di alta qualità, provenienti da startup AI leader e da Amazon stessa, disponibili attraverso un'API unificata. Questo servizio consente di scegliere il modello più adatto alle specifiche esigenze di un caso d'uso e di creare applicazioni di intelligenza artificiale generativa con elevati standard di sicurezza, privacy e responsabilità.

Con Amazon Bedrock, è possibile personalizzare privatamente i FM utilizzando tecniche come il fine-tuning e il *Retrieval Augmented Generation* (RAG), integrandoli facilmente nelle applicazioni senza dover gestire infrastrutture. Tra i modelli disponibili vi è Claude di Anthropic, un modello avanzato per la generazione di testo. Amazon Bedrock supporta anche la creazione di agenti in grado di eseguire compiti utilizzando sistemi e fonti di dati aziendali, migliorando l'efficienza e la precisione delle applicazioni basate su Generative AI.

Nel contesto del presente progetto, Amazon Bedrock e in particolare il modello Claude non sono stati utilizzati direttamente, in quanto si è ritenuto l'utilizzo di Amazon Comprehend e Amazon Textract sufficiente per le esigenze di analisi del testo e dei documenti.

Figura 3.8: Logo di Amazon Bedrock

3.2 Strumenti di sviluppo

Nel corso del progetto sono stati impiegati diversi strumenti di sviluppo che hanno contribuito in modo significativo alla realizzazione dell'applicazione. Tali strumenti hanno facilitato la scrittura, il testing e il monitoraggio del codice, consentendo una gestione efficiente del ciclo di sviluppo. Di seguito vengono descritti i principali strumenti utilizzati.

3.2.1 Jupyter Notebook

Jupyter Notebook (logo riportato in Figura 3.9) è un'applicazione web open-source che consente di creare e condividere documenti interattivi contenenti codice eseguibile, testo descrittivo, grafici e altri elementi multimediali. Jupyter supporta una vasta gamma di linguaggi di programmazione, tra cui Python, R e Julia, ed è ampiamente utilizzato in ambiti di ricerca, analisi dati e prototipazione di modelli di machine learning (ML).

In questo progetto, Jupyter Notebook ha svolto un ruolo centrale nella fase di prototipazione, in quanto è stato utilizzato per eseguire analisi esplorative dei dati, testare le funzionalità di *Amazon Com-*

prehend e *Amazon Textract*, e sviluppare i modelli di classificazione. Grazie alla sua natura interattiva, Jupyter ha consentito un rapido ciclo di test e iterazione, migliorando l'efficienza complessiva durante lo sviluppo dei modelli.

Figura 3.9: Logo di Jupyter Notebook

3.2.2 Visual Studio Code

Visual Studio Code (logo riportato in Figura 3.10) è un editor di codice sorgente sviluppato da Microsoft, disponibile per diversi sistemi operativi tra cui Windows, Linux e macOS. Si distingue per la sua leggerezza, la versatilità e l'ampia gamma di estensioni, che ne permettono l'integrazione con molteplici strumenti e linguaggi di programmazione.

Nel contesto del progetto, Visual Studio Code è stato utilizzato per sviluppare il codice dell'applicazione, inclusi i *Lambda functions* e i notebook Python. Inoltre, è stato impiegato per redigere e mantenere la documentazione tecnica del progetto, grazie alla sua integrazione con sistemi di controllo di versione come Git. Le sue funzionalità avanzate, come il supporto per il debug, la gestione delle estensioni per diversi linguaggi e l'integrazione con AWS, hanno contribuito a semplificare lo sviluppo e la gestione del progetto.

Figura 3.10: Logo di Visual Studio Code

3.2.3 Git

Git (logo riportato in Figura 3.11) è uno dei più popolari sistemi di controllo di versione distribuiti (Version Control System), utilizzato ampiamente nel settore dello sviluppo software per monitorare e gestire le modifiche al codice sorgente. Git permette a più sviluppatori di collaborare su un progetto, tenendo traccia delle modifiche, gestendo versioni multiple del software e consentendo il ripristino di versioni precedenti.

Nel progetto, Git è stato utilizzato per tracciare tutte le modifiche al codice sorgente, garantendo la gestione delle versioni e permettendo il lavoro collaborativo. Grazie alle sue funzionalità di branching e merging, Git ha facilitato lo sviluppo parallelo e la gestione dei vari task implementativi.

Figura 3.11: Logo di Git

3.2.4 Bitbucket

Bitbucket (logo riportato in Figura 3.12) è un servizio di hosting di repository Git basato su cloud, sviluppato da Atlassian. Oltre a supportare Git, Bitbucket offre integrazioni con strumenti di gestione dei progetti come Jira e Trello, rendendolo particolarmente adatto per team di sviluppo che seguono metodologie Agile.

All'interno del progetto, Bitbucket è stato utilizzato per ospitare il codice sorgente, fornendo un ambiente centralizzato e sicuro per la gestione del repository Git. Le funzionalità di collaborazione, come la revisione del codice e la gestione dei pull request, hanno permesso un efficace controllo della qualità del codice sviluppato.

Figura 3.12: Logo di Bitbucket

3.2.5 Python

Python (logo riportato in Figura 3.13) è un linguaggio di programmazione ad alto livello, interpretato, noto per la sua semplicità sintattica e la vasta libreria di moduli disponibili, che ne fanno una scelta eccellente per un'ampia gamma di applicazioni, tra cui sviluppo web, desktop, scientifico e AI.

Nel presente progetto, Python è stato il linguaggio di riferimento per la realizzazione delle *Lambda functions* utilizzate su AWS e per lo sviluppo dei notebook di Jupyter. La sua ampia compatibilità con le librerie di ML, come *TensorFlow*, *scikit-learn* e *Keras*, ha reso Python lo strumento ideale per lo sviluppo e l'addestramento dei modelli di apprendimento automatico impiegati nel progetto.

Figura 3.13: Logo di Python

Capitolo 4

Progettazione e codifica

In questo capitolo si descrive la progettazione e la codifica del sistema. Si inizia con una panoramica generale del sistema, per poi passare a una descrizione dettagliata delle varie componenti.

4.1 Architettura ad alto livello

Le fasi di un flusso di lavoro per l'IDP possono variare in base al caso d'uso specifico e ai requisiti aziendali, ma esistono alcune fasi comuni che sono generalmente presenti in qualsiasi processo IDP. Tali flussi di lavoro trovano applicazione in diversi ambiti, come l'elaborazione di moduli fiscali, reclami, note mediche, moduli di nuovi clienti, fatture, contratti legali, e molti altri documenti aziendali.

Nel contesto del presente progetto, l'obiettivo è stato quello di rispondere alla richiesta dell'azienda ospitante di automatizzare la catalogazione e l'elaborazione delle email e dei relativi documenti allegati. A tal fine, è stato progettato un flusso di lavoro articolato in diverse fasi, ciascuna delle quali contribuisce a trasformare i documenti non strutturati in informazioni strutturate e utilizzabili. Le fasi individuate per il processo di elaborazione dei documenti dalle email sono le seguenti:

- ***Data Capture***: Questa fase riguarda l'estrazione degli allegati dalle email. I file vengono archiviati e aggregati in modo sicuro, garantendo la corretta gestione dei dati fin dal primo momento. Questo passaggio è cruciale per assicurare che tutte le informazioni necessarie siano raccolte e pronte per le fasi successive del processo.

- ***Classification***: Una volta acquisiti, i documenti vengono classificati in base al loro contenuto. Questa fase consiste nell'assegnazione di ciascun documento a una specifica pipeline di elaborazione, in base alla tipologia di documento identificata. La corretta classificazione è fondamentale per assicurare che ogni documento segua il percorso di elaborazione più appropriato.

- ***Extraction***: Durante questa fase, vengono estratte le informazioni aziendali rilevanti dai documenti. Si tratta di un processo automatizzato in cui i dati chiave vengono isolati e resi disponibili per ulteriori analisi. L'accuratezza di questa fase è determinante per il successo complessivo del flusso di lavoro, poiché influisce direttamente sulla qualità delle informazioni che verranno utilizzate.

- ***Validation***: Una volta estratte, le informazioni devono essere validate. In questa fase, vengono applicate regole di business per assicurare che i dati siano corretti e completi. Inoltre viene controllata la confidenza per ogni informazione estratta, riducendo il margine di errore e assicurando l'affidabilità del processo.

- ***Storage***: Infine, le informazioni validate vengono salvate in un database aziendale. Questo passaggio è essenziale per garantire che i dati estratti siano facilmente accessibili per future consultazioni o analisi, completando così il ciclo di trasformazione dei documenti.

Questo flusso di lavoro, progettato per ottimizzare l'elaborazione automatizzata dei documenti, rappresenta un passo significativo verso l'efficienza operativa e la riduzione dei costi aziendali. Tale flusso è stato implementato utilizzando i servizi di AWS, in particolare *AWS Lambda*, *Amazon Textract*, *Amazon Comprehend* e *Amazon DynamoDB*. Attraverso *AWS Step Functions* è stato possibile orchestrare in modo efficiente le diverse fasi del processo, garantendo una gestione ottimale dei dati e una maggiore scalabilità.
L'architettura proposta (figura 4.1) è stata concepita all'interno del cloud AWS affidato da Sanmarco Informatica S.p.A. , in particolare nel portale denominato *WikiAi*. Le risorse principali sono state concepite nella regione Francoforte (eu-central-1) e sono state organizzate in base alle esigenze del progetto. Come precedentemente menzionato l'architettura comprende diverse fasi, ognuna delle quali svolge un ruolo specifico nel processo di elaborazione dei documenti. In particolare, il flusso di lavoro è stato progettato per classificare gli allegati delle email in quattro categorie principali: ordini, fatture, contratti e non classificato. Inoltre, il sistema è progettato per estrarre informazioni specifiche dai documenti appartenenti alle prime tre categorie, escludendo la categoria non classificato.

Come precedentemente menzionato, gran parte del flusso di lavoro è stato orchestrato tramite AWS Step Functions (nella figura 4.1 è evidenziato tramite un riquadro in rosso) e in particolare tramite la state machine denominata *IdpStateMachine*. La figura 4.2, creata tramite l'editor di AWS, mostra la struttura della state machine e le diverse fasi coinvolte nel processo di elaborazione dei documenti.

4.2 Risorse e servizi AWS utilizzati

Il sistema è stato progettato utilizzando una serie di risorse e servizi AWS, ciascuno dei quali svolge un ruolo specifico nel processo di elaborazione dei documenti. Le principali risorse e servizi utilizzati includono:

- Amazon S3
  - *S3 Emails Bucket*: contenente le email in formato `.eml`.
  - *S3 Attachments Bucket*: contenente gli allegati estratti dalle email.
  - *S3 First Pages Attachments Bucket*: contenente le prime pagine dei file PDF estratte dagli allegati.

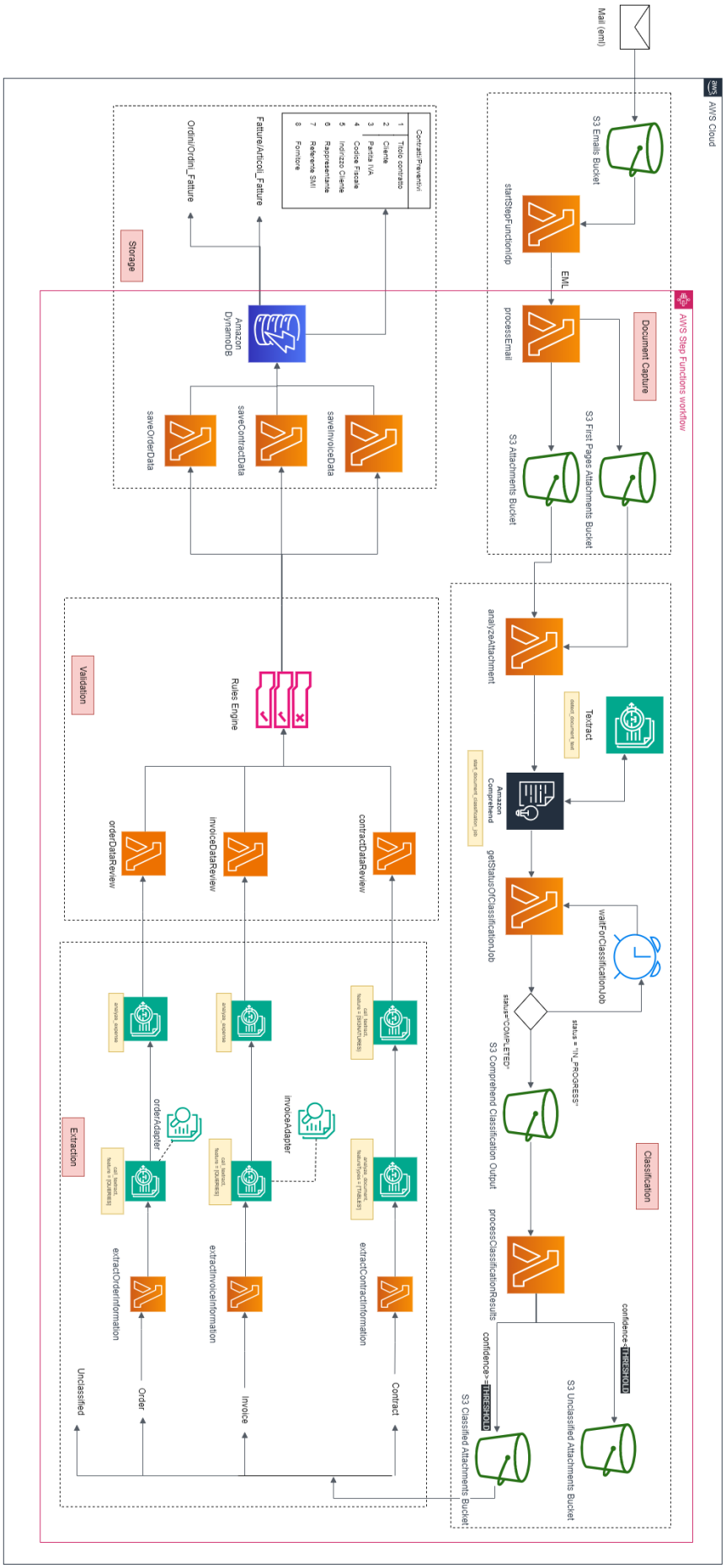


Figura 4.1: Architettura ad alto livello del sistema

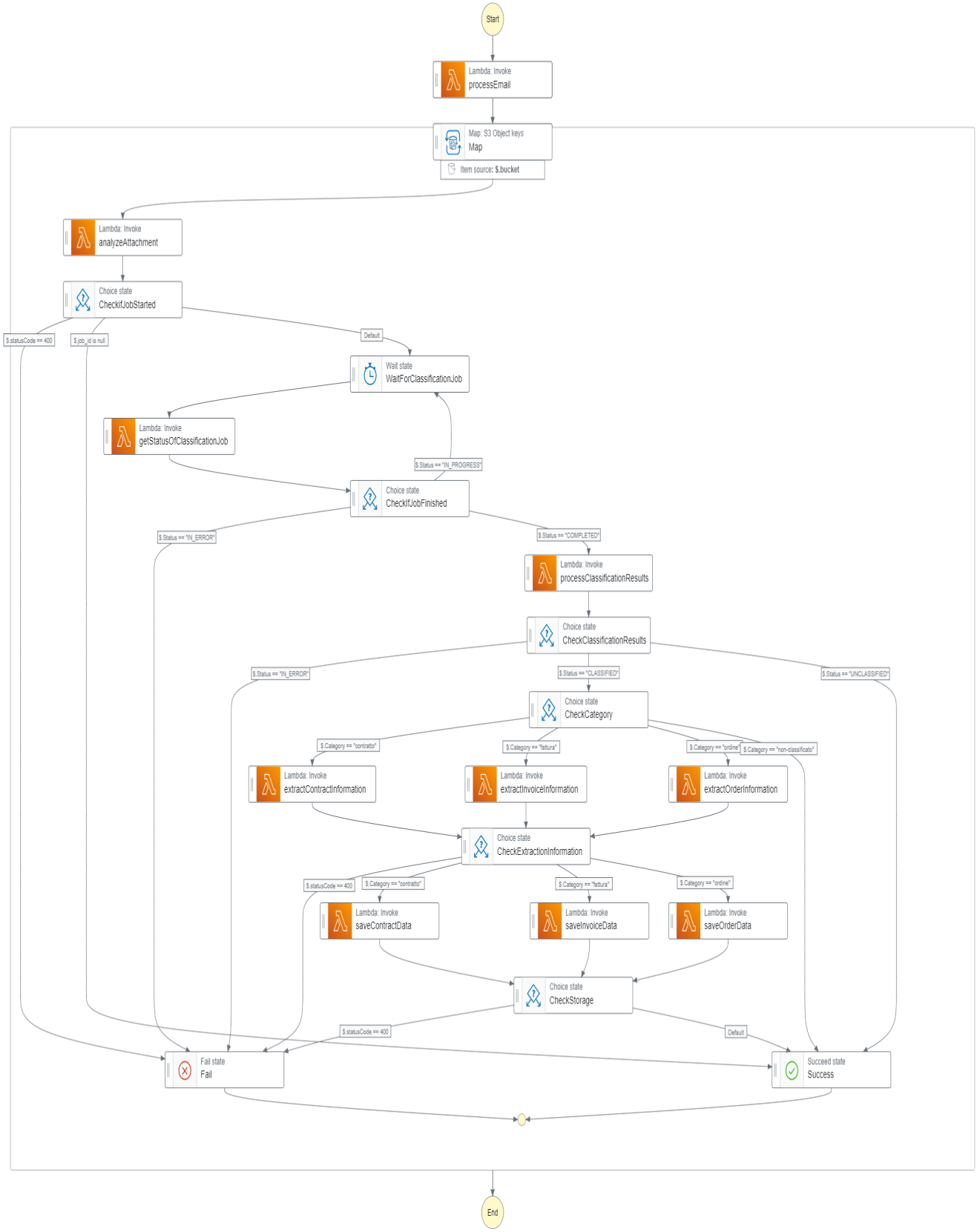


Figura 4.2: State machine "IdpStateMachine" di AWS Step Functions

  - *S3 Comprehend Classification Output*: contenente i risultati della classificazione di Comprehend.
  - *S3 Classified Attachments Bucket*: contenente gli allegati classificati.
  - *S3 Unclassified Attachments Bucket*: contenente gli allegati non classificati.

- AWS Lambda
  - *startStepFunctionIdp*: attiva l'esecuzione della state machine IdpStateMachine.
  - *processEmail*: estrae gli allegati dalle email.
  - *analyzeAttachment*: utilizza il classificatore di Comprehend per classificare gli allegati.
  - *getStatusOfClassificationJob*: controlla lo stato del job di classificazione.
  - *processClassificationResults*: elabora l'output della classificazione di Comprehend.
  - *extractContractInformation*: estrae le informazioni dai contratti.
  - *extractInvoiceInformation*: estrae le informazioni dalle fatture.
  - *extractOrderInformation*: estrae le informazioni dagli ordini.
  - *contractDataReview*: permette la revisione manuale delle informazioni estratte dai contratti.
  - *invoiceDataReview*: permette la revisione manuale delle informazioni estratte dalle fatture.
  - *orderDataReview*: permette la revisione manuale delle informazioni estratte dagli ordini.
  - *saveContractInformation*: salva le informazioni estratte dai contratti in DynamoDB.
  - *saveInvoiceInformation*: salva le informazioni estratte dalle fatture in DynamoDB.
  - *saveOrderInformation*: salva le informazioni estratte dagli ordini in DynamoDB.

- AWS Step Functions
  - *IdpStateMachine*: gestisce il flusso di lavoro.

- Amazon Comprehend
  - *document-classifier*: modello personalizzato per la classificazione degli allegati.
  - *custom-document-classifier-flywheel*: flywheel per la creazione di modelli personalizzati che riporta tre differenti dataset:
    * *document-classifier-train*: dataset di training per la prima versione del modello.
    * *trainingFatture*: dataset di training per la seconda versione del modello.
    * *my-training-set*: dataset di training per la seconda versione del modello.

- Amazon Textract
  - *analyze_document*: estrae le informazioni principali dai documenti.
  - *call_textract*: funzione simile a *analyze_document* ma con funzionalità aggiuntive.

  - *adapter checksInvoiceAdapter*: adattatore utilizzato per le custom queries per l'estrazione delle informazioni dalle fatture.
  - *adapter checksOrderAdapter*: adattatore utilizzato per le custom queries per l'estrazione delle informazioni dagli ordini.

- Amazon DynamoDB
  - *Contratti*: tabella contenente le informazioni estratte dai contratti.
  - *Preventivi*: tabella contenente le informazioni estratte dai preventivi.
  - *Fatture*: tabella contenente le informazioni estratte dalle fatture.
  - *Articoli_Fatture*: tabella contenente le informazioni relative agli articoli delle fatture.
  - *Ordini*: tabella contenente le informazioni estratte dagli ordini.
  - *Articoli_Ordini*: tabella contenente le informazioni relative agli articoli degli ordini.

4.3 Estrazione degli allegati

Inizialmente, l'indicazione fornita dall'azienda richiedeva un'analisi del contenuto delle email, seguita da una classificazione basata sull'elaborazione del linguaggio naturale e sui metadati contenuti. Tuttavia, con il chiarimento delle categorie di interesse (ordini, fatture e contratti), ho deciso di concentrare l'attenzione sull'estrazione degli allegati presenti nelle email piuttosto che sul contenuto testuale delle stesse.

Questa scelta è giustificata dal fatto che i documenti di interesse per l'azienda sono spesso inclusi come allegati nelle email, rendendo l'estrazione degli allegati un approccio più diretto ed efficace. Inoltre, il contenuto delle email è spesso irrilevante o solo parzialmente utile per il processo di classificazione.

L'obiettivo principale di questa fase è quindi quello di ricavare gli allegati dalle email per poterli successivamente classificare e analizzare. Il processo è strutturato nelle seguenti fasi:

- **Caricamento del file `.eml`**: Il processo inizia con il caricamento del file `.eml` nel bucket "S3 Emails Bucket", che funge da archivio per le email da analizzare.
- **Attivazione della funzione Lambda "StartStepFunctionIdp"**: L'inserimento del file `.eml` nel bucket attiva la funzione Lambda *StartStepFunctionIdp*, la quale avvia l'esecuzione della state machine *IdpStateMachine* di AWS *Step Functions*. Questa state machine gestisce l'intero flusso di lavoro automatizzato.
- **Estrazione degli allegati**: La state machine avvia la funzione Lambda *processEmail*, responsabile dell'estrazione degli allegati dalle email. Questa funzione è essenziale per isolare i documenti di interesse dal file `.eml`.
- **Caricamento degli allegati**: Una volta estratti, gli allegati vengono caricati nel bucket *S3 Attachments Bucket*. Gli allegati, che possono essere in vari formati (PDF, PNG, JPG, TXT, DOC,

DOCX, ecc.), vengono archiviati in una cartella il cui nome corrisponde a quello della mail da cui provengono (file `.eml`). In aggiunta, le prime pagine dei file PDF vengono salvate nel bucket *S3 First Pages Attachments Bucket* e vengono archiviati nello stesso modo.

Gli allegati, ora presenti nei bucket *S3 Attachments Bucket* e *S3 First Pages Attachments Bucket*, sono pronti per essere classificati e analizzati nelle fasi successive del processo.

4.4 Classificazione dei documenti

Inizialmente, si era presa in considerazione l'idea di classificare le email in base al loro contenuto utilizzando modelli di ML offerti da *Amazon SageMaker*. Tuttavia, con il chiarimento delle categorie di interesse durante lo stage (ordini, fatture e contratti), si è deciso di focalizzarsi sull'estrazione e la classificazione degli allegati presenti nelle email, piuttosto che sul contenuto testuale delle stesse. Questo approccio ha semplificato significativamente il processo di classificazione, poiché i metadati (denominati anche *features* nel contesto del ML) si riducono al semplice testo estratto dagli allegati e tale caso d'uso si presta bene all'utilizzo di *Amazon Comprehend* per la classificazione ed in particolare per la creazione di un modello personalizzato.
Sebbene in una fase iniziale fosse stato considerato e provato l'uso del servizio *Amazon Bedrock* per classificare i documenti, in particolare con il modello *Claude-3*, questa opzione è stata successivamente scartata a favore di un modello personalizzato, ritenuto più adatto alle specifiche esigenze del progetto e meno costoso.
In questo contesto, l'utilizzo di *Amazon Textract* e *Amazon Comprehend* si è rivelato cruciale per l'accuratezza della classificazione.
La fase di classificazione degli allegati (figura 4.1 riquadro denominato *Classification*) si articola nelle seguenti operazioni:

- **Caricamento del file `.eml` e attivazione della pipeline**: Dopo l'estrazione degli allegati dalle email, descritta nella Sezione 4.3, i documenti vengono preparati per la classificazione. La funzione Lambda *analyzeAttachment* utilizza il classificatore di Amazon Comprehend denominato *document-classifier* per analizzare la prima pagina degli allegati ricavandole dal bucket *S3 First Pages Attachments Bucket*. Il testo viene estratto tramite la funzione `detect_document_text` di *Amazon Textract*, che converte i contenuti dei documenti in un formato testuale adatto per l'analisi.

- **Salvataggio del risultato della classificazione**: I risultati del job di classificazione, composti da un file JSON che riporta la categoria assegnata e il relativo livello di confidenza, vengono salvati nel bucket *S3 Comprehend Classification Output*. Questo passaggio consente di archiviare in modo strutturato i risultati che saranno successivamente elaborati per l'estrazione delle informazioni.

- **Processamento dei risultati della classificazione**: Al termine del job di classificazione, la funzione Lambda *processClassificationResults* viene attivata per salvare gli allegati classificati nei *bucket* appropriati. Infatti se la confidenza del modello è superiore a una soglia (*threshold*) predefinita, l'allegato viene salvato nel *bucket S3 Classified Attachments Bucket*; altrimenti, viene

salvato nel *bucket S3 Unclassified Attachments Bucket*. Gli allegati sono archiviati in cartelle che riportano la categoria di classificazione (ordine, fattura, contratto, non classificato). Questa organizzazione è fondamentale per facilitare la gestione e l'analisi successiva degli allegati, compresi quelli non classificati, che possono essere esaminati in dettaglio in un secondo momento.

Una precisazione importante riguarda la distinzione tra gli allegati salvati nel bucket *S3 Classified Attachments Bucket* e quelli salvati nel bucket *S3 Unclassified Attachments Bucket* all'interno della cartella "non classificato". Gli allegati presenti nel bucket *S3 Unclassified Attachments Bucket* sono quelli per cui il livello di confidenza del modello è inferiore alla soglia predefinita. Al contrario, gli allegati presenti nella cartella "non classificato" del *S3 Classified Attachments Bucket* sono stati classificati con una confidenza superiore alla soglia, ma la categoria assegnata è comunque "non classificato", indicando la classificazione sia stata effettuata con un certo grado di sicurezza.

4.4.1 Creazione del modello di classificazione personalizzato

La creazione di un modello di classificazione personalizzato con Amazon Comprehend richiede la disponibilità di un dataset ampio, significativo e bilanciato, capace di distinguere con precisione le categorie di interesse. È fondamentale che il dataset sia etichettato correttamente, in modo che ogni documento sia associato alla giusta categoria di appartenenza.

Durante la fase di etichettatura, sono emerse alcune considerazioni chiave. I documenti da analizzare sono prevalentemente file PDF, spesso costituiti da scansioni. Per garantire coerenza nel processo di training, si è deciso di utilizzare esclusivamente documenti in formato PDF. Inoltre, è stato scelto di utilizzare unicamente le prime pagine di tali documenti per il training del modello. Questa scelta è stata motivata dal fatto che le prime pagine contengono generalmente le informazioni più rilevanti per la classificazione. Inoltre, considerando che il costo dell'analisi è proporzionale al numero di pagine, la riduzione del numero di pagine ha comportato una significativa riduzione dei costi operativi, soprattutto in documenti che possono arrivare fino a 30 o più pagine.

Tuttavia, queste scelte hanno anche portato a una riduzione della varietà dei dati utilizzati per il training, il che potrebbe potenzialmente introdurre [g]bias nel modello, limitando la sua capacità di generalizzare su nuovi dati.

Del modello personalizzato denominato *document-classifier* sono state create due versioni, ciascuna addestrata su dei dataset specifici. Per la prima versione del modello, denominata *document-classifier-version-1*, sono stati utilizzati 47 documenti etichettati.

Per la seconda versione del modello, denominata *Comprehend-Generated-v1- 461f932*, generata tramite il processo di active learning con *Flywheel*, sono stati utilizzati dei dataset di training di 56 documenti che si vanno ad aggiungere ai 47 documenti utilizzati per la versione precedente. Per creare e distribuire il modello personalizzato, sono state seguite le seguenti fasi: *Analisi del dataset*, *Preprocessing*, *Training*, *Valutazione* e *Test del modello*.

Analisi del dataset

L'*analisi del dataset* è fondamentale per comprendere la distribuzione delle categorie e valutare l'adeguatezza dei dati per la successiva fase di addestramento. Per i dataset utilizzati nelle varie iterazioni, le percentuali di distribuzione delle categorie (ordini, fatture, contratti e non classificato) sono state attentamente monitorate per garantire un bilanciamento adeguato.
Per il primo dataset di training, denominato *document-classifier-train*, sono stati utilizzati 47 documenti etichettati, distribuiti nel modo seguente:

- 25 contratti (53.19%)
- 3 fatture (6.38%)
- 5 ordini (10.64%)
- 14 non classificati (29.79%)

Per il secondo dataset di training, denominato *trainingFatture*, sono stati utilizzati 26 documenti etichettati, distribuiti nel modo seguente:

- 1 ordine (3.85%)
- 25 fatture (96.15%)

Per il terzo dataset di training, denominato *my-training-set*, sono stati utilizzati 30 documenti etichettati, distribuiti nel modo seguente:

- 10 fatture (33.33%)
- 10 non classificati (33.33%)
- 10 ordini (33.33%)

Tali scelte sono state guidate dalla necessità di garantire un bilanciamento adeguato delle categorie, in modo da evitare che il modello sia influenzato da una distribuzione sbilanciata dei dati. Inoltre, è stato fondamentale assicurare che i documenti etichettati fossero rappresentativi delle categorie di interesse, in modo da garantire che il modello fosse in grado di generalizzare su nuovi dati.

Preprocessing

Il preprocessing dei dati è una fase critica del processo di addestramento. Le operazioni principali sono state:

- **Estrazione del testo tramite Amazon Textract**: Il testo contenuto nelle prime pagine dei PDF è stato estratto utilizzando *Amazon Textract* tramite l'API `call_textract`.
- **Creazione del file CSV**: I dati estratti sono stati organizzati in un file CSV, con una colonna per la categoria di classificazione e una per il testo.
- **Caricamento del file CSV**: Il file di training CSV è stato caricato su *Amazon S3* tramite *Flywheel*, per essere utilizzato nel processo di *training*.

Training

Durante la fase di training, il file CSV creato in precedenza è stato utilizzato per addestrare una nuova versione del classificatore personalizzato all'interno di *Amazon Comprehend*. Il processo di training è stato eseguito utilizzando il servizio *Flywheel*, che consente di creare e gestire modelli personalizzati in modo efficiente. Il modello è stato addestrato su un'istanza *ml.m5.xlarge* per garantire prestazioni ottimali e tempi di risposta rapidi.

Valutazione

La valutazione del modello è stata condotta utilizzando metriche standard, che hanno riportato i seguenti risultati per entrambe le versioni del modello:

- Precision: 1.0
- Recall: 1.0
- F1: 1.0
- Accuracy: 1.0
- Micro precision: 1.0
- Micro recall: 1.0
- Micro F1: 1.0

Questi risultati indicano una performance ottimale del modello sulle classi di interesse.
TO DO: spiegare meglio le metriche di valutazione.

Test del modello

Per testare il modello, è sufficiente caricare il file desiderato in un *bucket S3* e avviare un *job* di classificazione. Il modello restituirà la categoria di classificazione assegnata al documento e il livello di confidenza associato, permettendo così una verifica immediata delle sue prestazioni.

4.4.2 Processo di Active Learning con Flywheel

Per migliorare il modello nel tempo, è stato utilizzato il processo di [g]active learning implementato tramite il servizio *Flywheel* di *Amazon Comprehend*. Questo approccio consente di iterare sul modello, migliorandolo progressivamente sulla base dei nuovi dati e delle prestazioni ottenute. Il processo segue questi passaggi:

- **Creazione di un *dataset Flywheel***: Si parte con la definizione di un *dataset* contenente i documenti etichettati, che verrà utilizzato per l'addestramento del modello.
- **Inizializzazione di un'iterazione *Flywheel***: Viene avviata un'iterazione di *Flywheel*, durante la quale il modello viene addestrato sui dati disponibili.

- Attivazione del nuovo modello: Sulla base dei risultati dell'iterazione, viene deciso se attivare il nuovo modello. La decisione si basa su parametri predefiniti, come le metriche di precisione, recall e F1 score.

4.5 Estrazione delle informazioni

In questa fase l'obiettivo è l'estrazione delle informazioni associate a ciascuna categoria escludendo la categoria non classificato. A partire dai risultati di classificazione della fase precedente si è analizzato il metodo migliore per poter estrarre le informazioni ricercate dalle categorie di contratti, ordini e fatture. Per ciascuna categoria utilizzata una funzione *lambda* che tramite *Amazon Textract* estrae le informazioni principali dai documenti. Per ciascuna delle informazioni estratte viene riportata anche la confidenza associata utile nella fase successiva.
Fondamentalmente sono stati analizzati diversi metodi utilizzando differenti servizi per aderire a tale scopo:

- Comprehend custom entities
- Amazon Bedrock con il modello Claude-3
- Servizi di Amazon Textract

Per ciascuna categoria è stata fatta dunque un'analisi che ha riguardato i costi oltre che l'efficacia del servizio.

4.5.1 Estrazione delle informazioni dai contratti

Per l'estrazione delle informazioni dai contratti, è stata adottata una soluzione efficace e a basso costo, sfruttando la struttura uniforme di questi documenti. I contratti analizzati presentano una tabella standardizzata che contiene le seguenti informazioni chiave:

- Titolo del contratto
- Cliente
- Partita IVA del cliente
- Codice fiscale del cliente
- Indirizzo del cliente
- Rappresentante legale del cliente
- Referente SMI
- Fornitore

La strategia implementata si basa sull'identificazione di questa tabella e sull'estrazione automatizzata dei campi utilizzando le conoscenze predefinite sulla struttura del documento. Per estrarre le informazioni desiderate, è stato impiegato *Amazon Textract*, utilizzando la funzione `analyze_document` con l'opzione `TABLES`, che consente di estrarre in modo efficiente i dati tabulari presenti nella prima pagina del contratto.

Per distinguere tra preventivi e contratti, è stata adottata una strategia basata sulla rilevazione delle firme all'interno del documento, sempre utilizzando *Amazon Textract*, ma con la funzione `SIGNATURES`. Il processo prevede l'analisi delle pagine a partire dall'ultima, alla ricerca di firme. Se una firma viene rilevata, il documento viene classificato come contratto. Se, invece, al termine dell'analisi di tutte le pagine, non viene trovata alcuna firma, il documento viene classificato come preventivo.

Il risultato di questa analisi viene salvato in un file JSON di output, includendo un campo `is_contract` che indica se il documento è stato classificato come contratto o preventivo. In caso di rilevamento di una firma, viene registrato anche il livello di confidenza associato. Se non viene trovata alcuna firma, la classificazione come preventivo viene effettuata con una confidenza del 100%. In entrambi i casi, viene applicata una soglia (*threshold*) di confidenza per garantire l'accuratezza del processo di rilevazione delle firme.

Questa soluzione permette di distinguere in modo efficace tra contratti e preventivi, sfruttando le funzionalità avanzate di *Amazon Textract* e mantenendo i costi operativi contenuti, senza compromettere l'affidabilità e la precisione del processo.

4.5.2 Estrazione delle informazioni dalle fatture

Per l'estrazione delle informazioni dalle fatture, sono state considerate diverse tecnologie, valutandone costi ed efficacia. Le soluzioni analizzate includono:

- Amazon Textract, in particolare la funzione `analyze_expense`
- Comprehend Custom Entity Recognition
- Amazon Bedrock con il modello Claude-3
- Utilizzo di *queries* e *custom queries* (adapters) di Amazon Textract

Per quanto riguarda l'analisi delle fatture tramite la funzione `analyze_expense` di Amazon Textract, questa è stata presa in considerazione poiché è specificamente progettata per l'analisi di fatture e ricevute, offrendo un'alta precisione per layout standard. Tuttavia, la sua efficacia è limitata a questi layout predefiniti e potrebbe non essere ottimale per fatture con strutture non convenzionali.

L'utilizzo di Comprehend Custom Entity Recognition consente di creare un modello personalizzato per l'estrazione di entità specifiche dalle fatture. Offre un'elevata precisione e flessibilità nell'adattarsi a layout variabili, ma presenta costi elevati e richiede un significativo sforzo nella preparazione e annotazione dei dati di addestramento.

Amazon Bedrock con il modello Claude-3 è stato valutato per la sua capacità di gestire layout complessi e variabili, garantendo un'elevata precisione. Tuttavia, il modello non è specificamente addestrato sui dati aziendali, il che potrebbe ridurre la sua efficacia per esigenze particolari.

L'utilizzo di *queries* e *custom queries* tramite *adapters* di Amazon Textract è stato infine scelto per l'analisi delle fatture grazie alla sua elevata precisione e flessibilità nel gestire layout variabili. Questa soluzione offre un buon compromesso tra personalizzazione, costi e precisione, con tempi di addestramento del modello inferiori rispetto ad altri metodi.

La soluzione basata su *queries* è risultata la più conveniente, offrendo un valore elevato di precisione per un numero limitato di fatture. L'integrazione con gli *adapters*, in particolare l'uso dell'*adapter* "checksInvoiceAdapter", ha garantito una personalizzazione ottimale mantenendo i costi contenuti.

Le informazioni principali estratte dalle fatture con questa soluzione sono:

- Numero fattura
- Data fattura
- Venditore
- Prezzo totale
- Partita IVA del venditore
- Codice fiscale del venditore
- Imponibile
- Imposta

Per quanto riguarda l'estrazione delle informazioni relative agli articoli delle fatture, l'utilizzo della funzione `analyze_expense` di Amazon Textract è emerso come la soluzione più efficace. Questa tecnologia offre un'elevata precisione e flessibilità nel gestire layout variabili, risultando particolarmente adatta per l'estrazione delle seguenti informazioni:

- Codice articolo
- Descrizione articolo
- Valore unitario
- Quantità
- Unità di misura
- Sconto percentuale
- IVA
- Imponibile articolo

Questa combinazione di strumenti e tecnologie ha permesso di ottimizzare il processo di estrazione delle informazioni dalle fatture, garantendo un'elevata accuratezza e adattabilità a diverse tipologie di documenti.

4.5.3 Estrazione delle informazioni dagli ordini

In questa fase si possono applicare considerazioni simili a quelle discusse per l'estrazione delle informazioni dalle fatture (sezione 4.5.2), poiché sono state prese in considerazione le stesse tecnologie, con problematiche analoghe.

Per l'estrazione delle informazioni principali dagli ordini, i campi identificati sono i seguenti:

- Numero ordine
- Data ordine
- Venditore
- Prezzo totale
- Partita IVA del venditore
- Codice fiscale del venditore
- Imponibile
- Imposta

Queste informazioni vengono estratte dai documenti utilizzando le *queries* di Amazon Textract, integrate con *custom queries* (*adapters*) per adattarsi a layout variabili. L'*adapter* utilizzato in questo contesto è denominato "checksOrderAdapter".

Per quanto riguarda l'estrazione delle informazioni relative agli articoli contenuti negli ordini, l'utilizzo della funzione `analyze_expense` di Amazon Textract è emerso come la soluzione più efficace. Questa tecnologia offre un'elevata precisione e flessibilità, adattandosi a diversi layout. Le informazioni estratte per ciascun articolo sono le seguenti:

- Codice articolo
- Descrizione articolo
- Valore unitario
- Quantità
- Unità di misura
- Sconto percentuale
- IVA

- Imponibile articolo
- Data consegna

L'adozione di queste tecnologie ha permesso di ottimizzare il processo di estrazione delle informazioni dagli ordini, garantendo un'elevata accuratezza e adattabilità a diverse tipologie di documenti, analogamente a quanto avviene per le fatture.

4.5.4 Creazione degli adapter

Gli *adapter* sono stati creati per personalizzare le query di Amazon Textract, consentendo l'estrazione di informazioni specifiche dai documenti. Questi *adapter* sono stati utilizzati per l'estrazione delle informazioni sia dalle fatture che dagli ordini, e sono denominati rispettivamente *checksInvoiceAdapter* e *checksOrderAdapter*.

4.5.4.1 ChecksInvoiceAdapter

L'*adapter checksInvoiceAdapter* è stato sviluppato per l'estrazione delle informazioni dalle fatture. Le query personalizzate configurate in questo *adapter* sono state pensate per coprire tutte le informazioni rilevanti, tra cui date, importi, identificativi e dettagli degli articoli. Le query utilizzate includono:

- What is the invoice date or billing date?
- What is the invoice ID or billing number?
- What is the total tax amount?
- What is the receiver tax ID?
- What is the bill to name?
- What is the vendor name?
- What is the vendor VAT number?
- What is the vendor taxpayer ID?
- What is the subtotal?
- What is the tax?
- What is the total?
- What are the article codes? (Articolo in Italian)
- What are the descriptions of the items?
- What are the unit prices of the articles? (Valore in Italian)
- What are the quantities per item?

- What are the units of measure for the items?
- What are the discounts per item?
- What is the VAT rate per item? (IVA in Italian)
- What are the taxable amounts per article? (Imponibile in Italian)
- How many items are listed in this invoice?

L'*adapter*, attualmente alla sesta versione, è stato addestrato su un dataset di 21 fatture, di cui 16 utilizzate per il training e 5 per il test.

4.5.4.2 ChecksOrderAdapter

L'*adapter checksOrderAdapter* è stato creato per l'estrazione delle informazioni dagli ordini. Anche in questo caso, le query sono state personalizzate per garantire un'accurata estrazione dei dati rilevanti, come le date, gli importi e i dettagli degli articoli. Le query configurate includono:

- What is the order date?
- What is the order ID?
- What is the bill to name?
- What is the internal code of the order?
- What is the taxpayer ID?
- What is the item amount in this order?
- What is the net amount in this order?
- What is the total amount in this order?
- What are the article codes? (Articolo in Italian)
- What are the descriptions of the items?
- What are the quantities per item?
- What are the units of measure for the items?
- What are the discounts per item?
- What are the delivery dates for each item?

L'*adapter*, attualmente alla seconda versione, è stato addestrato su un dataset di 10 ordini, di cui 5 utilizzati per il training e 5 per il test.

Questi *adapter* hanno permesso di migliorare significativamente la precisione e l'efficacia dell'estrazione delle informazioni dalle fatture e dagli ordini, adattandosi ai layout variabili dei documenti e garantendo un'alta qualità dei dati estratti.

4.6 Validazione delle informazioni

In questa fase, le informazioni estratte dai documenti vengono sottoposte a un processo di validazione. Per ciascuna categoria di documento (ordine, fattura, contratto) è stata implementata una funzione Lambda specifica per la validazione dei dati. La strategia di validazione applicata varia in base alla categoria di appartenenza, con regole specifiche per ogni tipo di dato. Se una delle regole di validazione non viene rispettata, il dato viene considerato non valido.

4.6.1 Validazione dei contratti

Per i contratti, le seguenti regole di validazione sono state definite:

- **Titolo contratto**: obbligatorio
- **Cliente**: obbligatorio
- **is_contract**: obbligatorio e di tipo booleano
- **Partita IVA del cliente**: obbligatoria

Sono inoltre state stabilite soglie di confidenza per ciascuna delle informazioni estratte. Se la confidenza è inferiore alla soglia stabilita, il dato viene considerato non valido. Ad eccezione del "Titolo contratto" (per cui la soglia è 0.8), la soglia scelta per le altre informazioni è pari a 0.9.

4.6.2 Validazione delle fatture

Per le informazioni generali delle fatture, sono state definite le seguenti regole di validazione:

- **Numero fattura**: obbligatorio
- **Data fattura**: deve avere un formato valido
- **Imposta**: deve essere un numero
- **Imponibile**: deve essere un numero
- **Prezzo totale**: deve essere un numero
- **Partita IVA**: deve avere una lunghezza di 11 caratteri e deve essere un numero

Per le informazioni relative agli articoli delle fatture, sono state definite le seguenti regole di validazione:

- **Valore unitario**: deve essere un numero
- **Quantità**: deve essere un numero
- **IVA**: deve essere un numero
- **Imponibile articolo**: deve essere un numero

4.6.3 Validazione degli ordini

Per le informazioni generali degli ordini, le regole di validazione stabilite sono le seguenti:

- **Codice interno**: obbligatorio
- **Data ordine**: deve avere un formato valido
- **Prezzo totale**: deve essere un numero
- **Partita IVA**: deve avere una lunghezza di 11 caratteri

Per le informazioni relative agli articoli degli ordini, sono state stabilite le seguenti regole di validazione:

- **Valore unitario**: deve essere un numero
- **Quantità**: deve essere un numero
- **IVA**: deve essere un numero
- **Imponibile articolo**: deve essere un numero
- **Data consegna**: deve avere un formato valido

In tutte le categorie, la validazione delle informazioni è cruciale per garantire l'integrità e l'accuratezza dei dati processati. Le soglie di confidenza e le regole definite permettono di filtrare i dati non validi, migliorando l'affidabilità del sistema.

4.7 Persistenza dei dati

L'obiettivo di questa fase è garantire la persistenza dei dati estratti. La scelta per la memorizzazione dei risultati è ricaduta su Amazon DynamoDB, grazie alla sua scalabilità e affidabilità. Il flusso di lavoro prevede l'utilizzo di funzioni Lambda dedicate per ciascuna categoria di documenti (contratti, ordini, fatture), che si occupano di salvare i dati estratti nelle rispettive tabelle di DynamoDB. Per ciascuna categoria di documenti è stata creata una funzione Lambda specifica, che salva i risultati nelle seguenti tabelle di DynamoDB:

- **Contratti**: Salva i dati relativi ai contratti nella tabella `Contratti` o, in caso di documenti classificati come preventivi, nella tabella `Preventivi`. La distinzione viene effettuata sulla base della variabile `is_contract` presente nel file JSON di output.
- **Fatture**: Salva le informazioni generali delle fatture nella tabella `Fatture` e gli articoli associati nella tabella `Articoli_Fatture`.
- **Ordini**: Salva le informazioni generali degli ordini nella tabella `Ordini` e gli articoli associati nella tabella `Articoli_Ordini`.

4.7.1 Contratti

Le informazioni estratte dai contratti vengono salvate nella tabella `Contratti` di DynamoDB. Se il documento viene identificato come preventivo, i dati vengono invece memorizzati nella tabella `Preventivi`. Questa distinzione si basa sul valore della variabile `is_contract` ottenuta durante l'analisi.

4.7.2 Fatture

Le informazioni generali delle fatture vengono salvate nella tabella `Fatture` di DynamoDB. Inoltre, ogni articolo associato alla fattura viene memorizzato separatamente nella tabella `Articoli_Fatture`, assicurando una gestione dettagliata e strutturata delle informazioni.

4.7.3 Ordini

Le informazioni generali degli ordini vengono salvate nella tabella `Ordini` di DynamoDB. Similmente alle fatture, gli articoli associati agli ordini vengono memorizzati nella tabella `Articoli_Ordini`, permettendo di gestire in modo efficace i dettagli relativi a ciascun ordine.

Questo approccio assicura che tutti i dati estratti dai documenti siano memorizzati in modo organizzato e facilmente accessibile, sfruttando le potenzialità di DynamoDB per garantire performance elevate e affidabilità nel tempo.

4.8 Analisi dei costi

TO DO

Capitolo 5

Sviluppi futuri

In questa sezione vengono proposti alcuni sviluppi futuri del sistema.

5.1 Analisi del contenuto della mail

Per poter analizzare il contenuto della mail ed estrarre le informazioni associate si può modificare la funzione lambda *processEmail* in modo tale da estrarre il testo della mail e non solo gli allegati ed eventualmente caricarlo su un altro bucket S3.
Inoltre, si può implementare un modello di classificazione di Comprehend per classificare il testo della mail in base al contenuto analogamente a quanto fatto per gli allegati e successivamente estrarre le informazioni associate.

5.2 Aggiunta di nuove categorie

Si possono aggiungere nuove categorie di classificazione se necessario andando a modificare il modello di classificazione di Comprehend e in particolare il dataset fornito. Inoltre, si possono aggiungere nuove funzioni lambda per l'estrazione delle informazioni associate a ciascuna categoria.

5.3 Completamento delle informazioni

Si possono completare le informazioni mancanti e non estratte nello step 2: information extraction come ad esempio la data, l'importo, il mittente, il destinatario, ecc interrogando il database DynamoDB. Questo lavoro si può fare prima dello step 3: human review.

5.3.1 Active learning workflow per migliorare il modello di classificazione

Si può implementare un workflow di active learning per migliorare il modello di classificazione di Comprehend.

Per avere un riferimento dettagliato si può consultare il seguente link: Active learning workflow for Amazon Comprehend custom classification La figura 5.1 mostra un esempio di active learning workflow che utilizza flywheel per migliorare il modello di classificazione di Comprehend. Le sottosezioni succes-

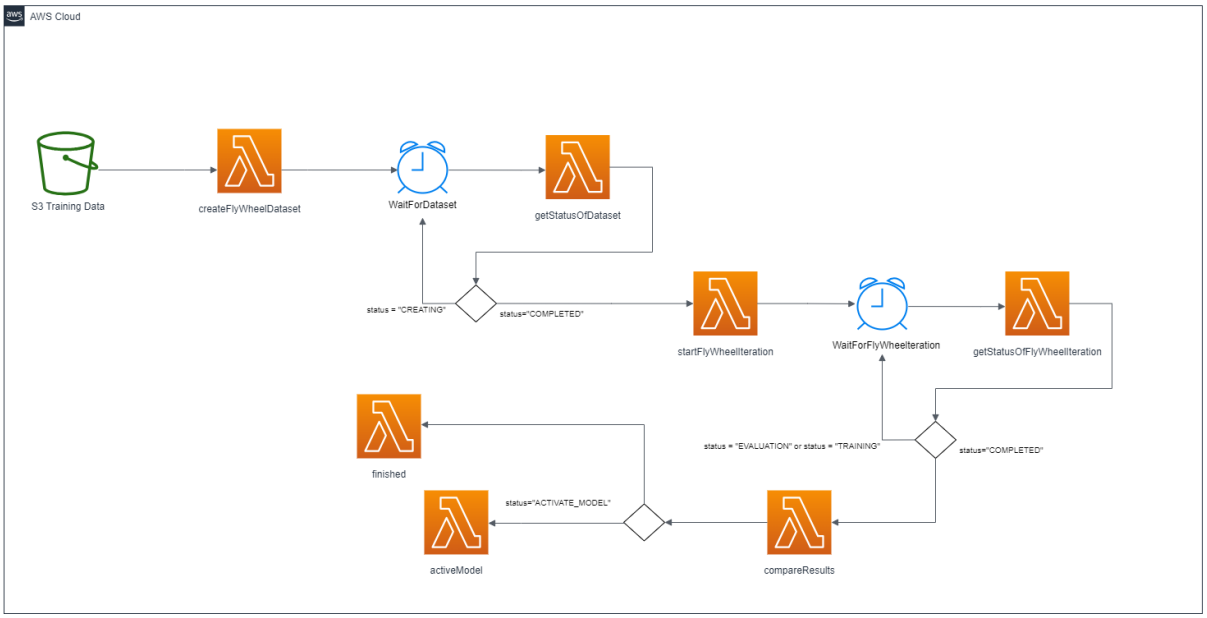


Figura 5.1: Active learning workflow

sive hanno lo scopo di fornire una panoramica generale su un esempio di active learning workflow per migliorare il modello di classificazione di Comprehend.

5.3.1.1 StartStepFunction

- **Trigger**: L'esecuzione è innescata dalla presenza di un file CSV contenente i dati di training.
- Il file CSV viene suddiviso in due file separati: uno per il training e uno per il testing.
- Viene avviata la Step Function.

5.3.1.2 StartCustomClassification

- Viene utilizzata la funzione `create_document_classifier` per creare un classificatore personalizzato.

Si attende un periodo di 10 secondi per consentire la creazione del classificatore.

5.3.1.3 GetStatusClassifier

- Viene utilizzata la funzione `describe_document_classifier` per ottenere lo stato attuale del classificatore. Gli stati possibili includono `SUBMITTED`, `TRAINING`, e altri stati come `IN_ERROR`, `TRAINED`, `DELETING`, e `CreateEndpoint`.
- Se la variabile `CurrentClassifierSSM` è impostata, viene restituito lo stato corrente del classificatore.

- Se la variabile `CurrentClassifierSSM` non è impostata:
  - Se lo stato è `TRAINED`, vengono salvate le variabili `CurrentClassifierSSM`, `CurrentTrainingDataSSM`, `CurrentTestDataSSM`, `CurrentTestingTruthDataSSM` e viene restituito lo stato `CreateEndpoint`.

Nel *choice state*, viene verificato:

- Se lo stato è `TRAINED`, si passa allo stato successivo `StartValidationTest`.
- Se lo stato è `CreateEndpoint`, si passa allo stato `StartCustomClassificationEndpointCreation`.
- In caso contrario, si attende per 10 secondi e si richiama `GetStatusClassifier`.

5.3.1.4 StartValidationTest

- Viene utilizzato il classificatore passato come evento e quello fornito come parametro.
- Viene eseguita la funzione `start_document_classification_job` su entrambi i modelli per classificare i dati di test.
- La risposta di entrambi i modelli viene stampata.
- Si passa allo stato successivo, trasmettendo l'ID di entrambi i job.

Si attende un periodo di 10 secondi per consentire la creazione dei job.

5.3.1.5 GetStatusValidationTest

- Viene utilizzata la funzione `describe_document_classification_job` per ottenere lo stato dei job dei due modelli. Gli stati possibili includono `COMPLETED`, `STOP_REQUESTED`, e altri stati come `STOPPED`, `FAILED`.

Nel *choice state*, si verifica se lo stato di entrambi i job è `COMPLETED`. In caso affermativo, si passa allo stato successivo; in caso contrario, si attende per 10 secondi e si richiama `GetStatusValidationTest`.

5.3.1.6 ComputeTestResults

- Vengono scaricati i risultati dei due job (file .gz).
- I risultati vengono salvati nella tabella `TestResultTable` su DynamoDB.
- Viene creata una matrice di confusione basata sui risultati e calcolati i parametri di *precision*, *recall*, e *f1-score*.
- I parametri vengono salvati nella tabella `TestResultTable`.
- Viene effettuato un confronto su un parametro selezionato:
  - Se il vecchio modello risulta migliore, viene restituito lo stato `DONT_CREATE`.

- In caso contrario, vengono aggiornate le variabili `CurrentClassifierSSM`, `CurrentTrainingDataSSM`, `CurrentTestDataSSM`, `CurrentTestingTruthDataSSM` con i valori del nuovo modello e viene restituito lo stato `CREATE`.

Se lo stato è `DONT_CREATE`, si passa alla lambda `FINISHED`; altrimenti, si procede alla lambda `StartCustomClassification`

5.4 Combinazione delle custom queries con analyze expense

Si può combinare l'uso delle custom queries con analyze expense per migliorare l'estrazione delle informazioni associate a ordini e fatture. In particolare, si può confrontare la percentuale di correttezza delle informazioni estratte con l'uso di queste due funzionalità.

5.5 Sviluppo di un'interfaccia grafica

Invece di caricare manualmente gli allegati o le email nel bucket S3, è possibile sviluppare un'interfaccia grafica che consenta l'upload diretto degli allegati e delle email. Questa soluzione semplificherebbe il processo, migliorando l'efficienza e l'usabilità del sistema.

5.6 Utilizzo di A2I (Amazon Augmented AI)

Per migliorare la precisione del modello, è possibile utilizzare Amazon Augmented AI (A2I). Maggiori informazioni sono disponibili al seguente link: Amazon Augmented AI.

5.7 Integrazione con un Sistema Documentale

È possibile integrare il sistema con un sistema documentale per l'archiviazione delle email e degli allegati, migliorando la gestione e l'accessibilità dei documenti.

5.8 Utilizzo di Amazon OpenSearch Service

Amazon OpenSearch Service può essere utilizzato per l'indicizzazione delle informazioni in Amazon DynamoDB, facilitando la ricerca delle informazioni. Questo servizio offre un'esperienza di ricerca sicura e scalabile.

5.9 Utilizzo di Amazon CloudFormation

Amazon CloudFormation può essere utilizzato per automatizzare la creazione e la gestione delle risorse AWS, semplificando il processo di provisioning dell'infrastruttura.

Capitolo 6

Conclusioni

In questa sezione vengono presentate le conclusioni del lavoro svolto.

Lorem [g]SDK

Lorem API

6.1 Consuntivo finale

Le ore di stage effettivamente svolte sono state 320, rispettando il monte ore previsto. Il lavoro svolto è stato suddiviso in diverse fasi, ognuna delle quali ha richiesto un impegno specifico. La fase iniziale di studio e formazione ha richiesto un tempo maggiore rispetto a quanto preventivato, in quanto ho dovuto approfondire le tecnologie e gli strumenti necessari per lo sviluppo del progetto. La fase di progettazione e sviluppo ha richiesto un impegno costante, ma sono riuscito a rispettare i tempi previsti. Infine, la fase di test e validazione ha richiesto un tempo inferiore rispetto a quanto preventivato, in quanto il prodotto sviluppato ha funzionato correttamente fin da subito.

6.2 Raggiungimento degli obiettivi

Gli obiettivi prefissati all'inizio del percorso (riportati nella sezione 2) di stage sono stati raggiunti con successo.

6.3 Conoscenze acquisite

TO DO

6.4 Valutazione personale

In conclusione, ritengo di essere soddisfatto del lavoro svolto e delle competenze acquisite durante il percorso di stage. L'esperienza ha rappresentato un'opportunità di crescita professionale e personale,

permettendomi di mettermi alla prova in un contesto lavorativo reale e di confrontarmi con problematiche complesse. Ho potuto apprendere concetti relativi al cloud computing e all'applicazione di servizi cloud, oltre che un particolare approfondimento sulle tecnologie di machine learning per l'elaborazione intelligente dei documenti. Inoltre, ho avuto modo di lavorare in un team di sviluppo, migliorando le mie capacità di collaborazione e di comunicazione. Infine, ho potuto mettere in pratica le competenze acquisite durante il percorso di studi, dimostrando di saper affrontare con successo le sfide che mi sono state proposte.

Acronimi e abbreviazioni

AI Artificial Intelligence. i, 5, 11, 13, 49

API Application Programming Interface. i, 49

AWS Amazon Web Services. i, 6, 13, 49

BPM Business Process Management. i

CPQ Configure, Price, Quote. i

ERP Enterprise Resource Planning. i

FM Foundation Models. i, 51

ICT Information and Communication Technology. i, 51

IDE Integrated Development Environment. i

IDP Intelligence Document Processing. i, 7, 51

LLM Large Language Model. i, 51

ML Machine Learning. i, 7, 14, 51

MLOps Machine Learning Operations. i, 14

NLP Natural Language Processing. i, 8, 52

OCR Optical Character Recognition. i, 8, 14, 52

PEC Posta Elettronica Certificata. i, 52

RPA Robotic Process Automation. i, 8

SDK Software Development Kit. i, 53

UML Unified Modeling Language. i, 53

VCS Version Control System. i

Glossario

Active learning Nell'ambito del machine learning per active learning si intende una tecnica di apprendimento supervisionato in cui il modello di machine learning è in grado di selezionare autonomamente i campioni di addestramento più informativi da un pool di dati non etichettati. L'active learning consente di ridurre il costo dell'etichettatura dei dati e di migliorare le prestazioni del modello. i, 30, 32, 49

Agile Nell'ambito dell'ingegneria del software con il termine Agile si intende un insieme di metodi di sviluppo del software basati su processi iterativi e incrementali, dove i requisiti e le soluzioni si evolvono attraverso la collaborazione tra team auto-organizzati e interfunzionali. i, 4, 11, 49

AI Per artificial Intelligence (AI) si intende l'insieme di tecnologie e metodi che permettono ai computer di eseguire attività che richiedono intelligenza umana, come il riconoscimento di immagini, il riconoscimento vocale, la traduzione automatica, ecc. i, 6, 8–11, 20, 22, 48

API In informatica con il termine *API* si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. i, 8, 12, 14–18, 20, 31, 46, 48

AWS Amazon Web Services (AWS) è una piattaforma di servizi cloud che offre potenza di calcolo, storage di database, distribuzione di contenuti e altre funzionalità per aiutare le imprese a scalare e crescere. i, 6, 8–11, 18, 21, 22, 24, 48

Bias Nell'ambito del machine learning per bias si intende il fenomeno per cui un modello di machine learning è incline a fare previsioni errate a causa di dati di addestramento non rappresentativi o di un'architettura del modello sbagliata. Il bias può portare a discriminazioni e disuguaglianze, e può essere ridotto attraverso la raccolta di dati più rappresentativi e l'ottimizzazione dell'architettura del modello. i, 30, 49

BPM Il Business Process Management (BPM) è un approccio sistemico alla gestione dei processi aziendali che mira a migliorare l'efficienza, la qualità e l'agilità dell'azienda. Il BPM coinvolge l'analisi, la progettazione, l'automazione e il monitoraggio dei processi aziendali per garantire che siano allineati agli obiettivi aziendali e alle esigenze dei clienti. i, 3, 48, 49

Bucket Nel contesto di AWS, per bucket si intende un contenitore di oggetti che consente di archiviare e organizzare i dati in Amazon S3. Un bucket può contenere un numero illimitato di oggetti e può essere configurato con diverse opzioni di accesso e sicurezza. i, 50

Business Intelligence La Business Intelligence (BI) è un insieme di processi, strumenti e tecnologie che consentono alle aziende di raccogliere, analizzare e presentare informazioni aziendali per supportare la presa di decisioni informate. La BI si basa sull'analisi dei dati storici e in tempo reale per identificare tendenze, modelli e opportunità di business. i, 2, 50

Computer Vision La Computer Vision è un'area dell'intelligenza artificiale che si occupa di creare sistemi che possono interpretare e comprendere le immagini e i video in modo simile agli esseri umani. La Computer Vision è utilizzata in applicazioni di riconoscimento facciale, riconoscimento di oggetti, veicoli autonomi e altre applicazioni di visione artificiale. i, 14, 50

CPQ Il Configure, Price, Quote (CPQ) è un processo aziendale che consente alle aziende di configurare, quotare e vendere prodotti e servizi in modo rapido, accurato e redditizio. Il CPQ coinvolge la configurazione dei prodotti, la determinazione dei prezzi e la generazione di preventivi personalizzati per i clienti. i, 3, 48, 50

Customer Intelligence La Customer Intelligence è l'insieme di processi, strumenti e tecnologie che consentono alle aziende di raccogliere, analizzare e utilizzare informazioni sui clienti per migliorare la customer experience, aumentare la fedeltà dei clienti e massimizzare il valore del cliente. i, 2, 50

Cybersecurity La cybersecurity è l'insieme di pratiche e tecnologie utilizzate per proteggere sistemi, reti e dati da attacchi informatici, accessi non autorizzati, e altre minacce digitali.. i, 2, 50

Data protection Misure tecniche e organizzative che proteggono i dati personali e aziendali da accessi non autorizzati, modifiche, divulgazioni o distruzioni.. i, 2, 50

Dataset Un dataset è un insieme di dati strutturati o non strutturati che vengono utilizzati per addestrare e valutare i modelli di machine learning. I dataset possono contenere dati di testo, immagini, audio, video o altri tipi di dati, e possono essere etichettati o non etichettati. i, 14, 27, 30, 31, 50

Deep Learning Il Deep Learning è un'area dell'intelligenza artificiale che si occupa di creare modelli di machine learning basati su reti neurali profonde. Il Deep Learning è utilizzato in applicazioni di riconoscimento di immagini, riconoscimento vocale, traduzione automatica e altre applicazioni di elaborazione del linguaggio naturale. i, 14, 15, 50

ERP Enterprise Resource Planning (ERP) è un sistema di gestione aziendale che integra e automatizza i processi aziendali, come la contabilità, la gestione delle risorse umane, la produzione, la logistica, le vendite e il marketing. Un sistema ERP consente di migliorare l'efficienza, la produttività e la collaborazione all'interno dell'azienda. i, 2, 48, 50

FM I Foundation Models (FM) sono modelli di linguaggio basati su reti neurali profonde che sono stati addestrati su un vasto corpus di testo per generare testo naturale in modo coerente e convincente. I FM sono utilizzati in applicazioni di generazione di testo, traduzione automatica, riassunto automatico e altre applicazioni di elaborazione del linguaggio naturale. i, 20, 48

GDPR Il Regolamento Generale sulla Protezione dei Dati (GDPR) è una legge sulla privacy che regola la protezione dei dati personali dei cittadini dell'Unione Europea. Il GDPR è entrato in vigore il 25 maggio 2018 e stabilisce regole chiare per la raccolta, l'elaborazione e la conservazione dei dati personali. i, 2, 51

Generative AI La Generative AI è un'area dell'intelligenza artificiale che si occupa di creare nuovi contenuti, come immagini, testo, musica e video, utilizzando modelli di machine learning generativi. La Generative AI è utilizzata in applicazioni di creazione di contenuti, design assistito da computer, e generazione di arte e musica. i, 8, 20, 51

Governance aziendale La governance aziendale è il sistema di regole, processi e pratiche che guidano e controllano le attività e le decisioni all'interno di un'azienda. La governance aziendale si occupa di definire gli obiettivi, le politiche e le procedure dell'azienda, di monitorare le prestazioni e di garantire la conformità alle normative e agli standard. i, 2, 51

IBM Power IBM Power Systems sono una famiglia di server e processori sviluppati da IBM utilizzati in ambienti aziendali per applicazioni critiche. i, 2, 51

ICT Con il termine Information and Communication Technology (ICT) si intende l'insieme delle tecnologie informatiche e telematiche utilizzate per la gestione delle informazioni e la comunicazione. i, 6, 48

IDE Un Integrated Development Environment (IDE) è un software che fornisce un ambiente integrato per lo sviluppo di software, comprensivo di editor di codice, compilatore, debugger e altre funzionalità di sviluppo. Un IDE semplifica il processo di sviluppo del software e aumenta la produttività dei programmatori. i, 18, 48, 51

IDP Con il termine Intelligence document processing (IDP) si intende l'insieme di tecnologie che permettono di estrarre informazioni da documenti cartacei o digitali, elaborarle e trasformarle in dati strutturati. i, 7, 8, 13, 23, 48

LLM Un Large Language Model (LLM) è un modello di linguaggio basato su reti neurali profonde che è stato addestrato su un vasto corpus di testo per generare testo naturale in modo coerente e convincente. Gli LLM sono utilizzati in applicazioni di generazione di testo, traduzione automatica, riassunto automatico e altre applicazioni di elaborazione del linguaggio naturale. i, 8, 48

ML Per Machine Learning (ML) si intende un insieme di tecniche e algoritmi che permettono ai computer di apprendere dai dati e di migliorare le prestazioni in base all'esperienza, senza essere esplicita-

mente programmati. Il machine learning si basa su modelli statistici e matematici che permettono di fare previsioni o decisioni in base ai dati analizzati. i, 8, 13, 18–20, 22, 29, 48

MLOps MLOps è una pratica che combina i principi e le pratiche dell'ingegneria del software con quelli del machine learning per creare, implementare e gestire modelli di machine learning in modo efficiente ed efficace. MLOps coinvolge la collaborazione tra team di sviluppo, data science e operazioni per garantire che i modelli di machine learning siano scalabili, affidabili e sicuri. i, 48, 52

NLP Natural Language Processing (NLP) è un campo dell'intelligenza artificiale che si occupa di interazioni tra computer e linguaggio umano. L'obiettivo principale di NLP è consentire ai computer di comprendere, interpretare e generare il linguaggio umano in modo che possano effettivamente comunicare con gli esseri umani in modo naturale. i, 8, 13, 48

NoSQL Il NoSQL è un'approccio alla gestione dei dati che si basa su modelli di dati non relazionali, come i database di documenti, i database di colonne, i database di grafi e i database chiave-valore. Il NoSQL è utilizzato per gestire grandi volumi di dati non strutturati e semi-strutturati in modo flessibile ed efficiente. i, 17, 52

OCR Optical Character Recognition (OCR) è una tecnologia che permette di convertire diversi tipi di documenti cartacei o digitali in testo digitale, in modo che possano essere elaborati e analizzati da un computer. i, 8, 48

PEC La *Posta Elettronica Certificata* (PEC) è un servizio di posta elettronica che garantisce l'invio e la ricezione di messaggi di posta elettronica con valore legale equivalente a quello della raccomandata con avviso di ricevimento. i, 3, 5, 6, 9–12, 48

Performance Management Il Performance Management è un processo continuo di pianificazione, monitoraggio e valutazione delle prestazioni dei dipendenti per garantire che raggiungano gli obiettivi aziendali. Il Performance Management coinvolge la definizione degli obiettivi, la valutazione delle prestazioni, il feedback e lo sviluppo delle competenze. i, 2, 52

Repository Con il termine repository si intende un ambiente di archiviazione centralizzato in cui vengono conservati e gestiti i file di un progetto software. Il repository consente di tenere traccia delle modifiche apportate ai file, di collaborare con altri sviluppatori e di mantenere una cronologia delle versioni del software . i, 52

RPA La Robotic Process Automation (RPA) è una tecnologia che consente di automatizzare i processi aziendali ripetitivi e basati su regole utilizzando software robot. I robot software possono eseguire attività manuali, ripetitive e noiose in modo rapido, accurato e senza errori. i, 48, 52

Scalabilità In informatica, la scalabilità è la capacità di un sistema di crescere in dimensioni e complessità in modo lineare o sub-lineare rispetto all'aumento del carico di lavoro. i, 7, 52

Scrum In ingegneria del software, per Scrum si intende un framework agile per la gestione del ciclo di sviluppo del software. Scrum è caratterizzato da un approccio iterativo e incrementale, in cui il lavoro è organizzato in sprints di durata fissa, di solito di 2-4 settimane. Scrum prevede un team auto-organizzato e interfunzionale, che lavora in modo collaborativo per raggiungere gli obiettivi prefissati. i, 4, 11, 12, 53

SDK Un Software Development Kit (SDK) è un insieme di strumenti e librerie di sviluppo software che consentono ai programmatori di creare applicazioni per una piattaforma specifica, come un sistema operativo, un framework o un servizio cloud. i, 10, 46, 48

Serverless Per serverless si intende un modello di cloud computing in cui il fornitore di servizi cloud gestisce l'infrastruttura del server e le risorse di calcolo, e il cliente paga solo per il tempo di esecuzione delle funzioni. Il modello serverless consente di ridurre i costi e semplificare la gestione delle risorse, in quanto il cliente non deve preoccuparsi di configurare e mantenere i server. i, 13, 17, 18, 53

UML In ingegneria del software *Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. i, 48

Version Control System Un Version Control System (VCS) è un sistema che registra le modifiche apportate ai file di un progetto software nel tempo, in modo che sia possibile ripristinare versioni precedenti, confrontare le modifiche e collaborare con altri sviluppatori. I VCS sono utilizzati per tenere traccia delle modifiche al codice sorgente e coordinare il lavoro di sviluppo. i, 21, 48, 53

Bibliografia

Siti web consultati

Active learning worflow for Amazon Comprehend custom models - Part 1. URL: https://aws.amazon.com/it/blogs/machine-learning/active-learning-workflow-for-amazon-comprehend-custom-classification-part-1/.

Amazon Comprehend Document Classifier adds layout support for higher accuracy. URL: https://aws.amazon.com/it/blogs/machine-learning/amazon-comprehend-document-classifier-adds-layout-support-for-higher-accuracy/.

Amazon Comprehend Examples for Building Custom Classifier. URL: https://github.com/aws-samples/amazon-comprehend-examples/blob/master/building-custom-classifier/BuildingCustomClassifie ipynb.

Automatically extract text and structured data from documents with Amazon Textract. URL: https://aws.amazon.com/it/blogs/machine-learning/automatically-extract-text-and-structured-data-from-documents-with-amazon-textract/.

AWS. URL: https://aws.amazon.com/.

AWS Samples for Intelligent Document Processing. URL: https://github.com/aws-samples/aws-ai-intelligent-document-processing.

AWS SDK for Python (Boto3) Samples. URL: https://github.com/awsdocs/aws-doc-sdk-examples.

Build a receipt and invoice processing pipeline with Amazon Textract. URL: https://aws.amazon.com/it/blogs/machine-learning/build-a-receipt-and-invoice-processing-pipeline-with-amazon-textract/.

Intelligent document processing with Amazon Textract, Amazon Bedrock, and Langchain. URL: https://aws.amazon.com/it/blogs/machine-learning/intelligent-document-processing-with-amazon-textract-amazon-bedrock-and-langchain/.

Intelligent document processing with AWS AI services: Part 1. URL: https://aws.amazon.com/it/blogs/machine-learning/part-1-intelligent-document-processing-with-aws-ai-services/.

Intelligent document processing with AWS AI services: Part 2. URL: https://aws.amazon.com/it/blogs/machine-learning/part-2-intelligent-document-processing-with-aws-ai-services/.

Introducing one-step classification and entity recognition with Amazon Comprehend for intelligent document processing. URL: https://aws.amazon.com/it/blogs/machine-learning/introducing-one-step-classification-and-entity-recognition-with-amazon-comprehend-for-intelligent-document-processing/.

Introducing specialized support for extracting data from invoices and receipts using Amazon Textract. URL: https://aws.amazon.com/it/blogs/machine-learning/announcing-expanded-support-for-extracting-data-from-invoices-and-receipts-using-amazon-textract/.

Introducing the Amazon Comprehend Flywheel for MLOps. URL: https://aws.amazon.com/it/blogs/machine-learning/introducing-the-amazon-comprehend-flywheel-for-mlops/.

Manifesto Agile. URL: http://agilemanifesto.org/iso/it/.