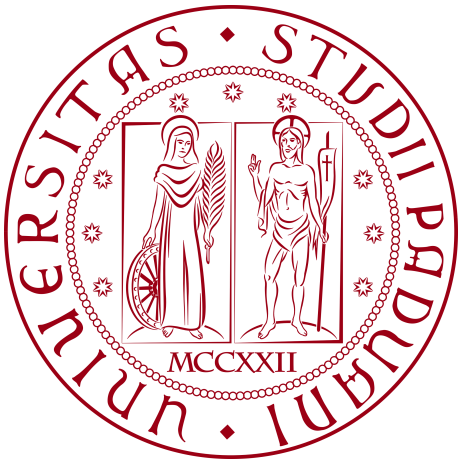
Università degli Studi di Padova

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

"I'm stronger, I'm smarter, I'm better"

— Homelander.

"Diventerò il re dei pirati"

— Monkey D. Luffy.

"Non devo fuggire"

— Shinji Ikari.

"Non dire gatto se non ce l'hai nel sacco"

— Nonna.

"GG\MM\AAAAAAA"

— Alex Scantamburlo.

"Ucciderò tutti i giganti"

— Eren Yeager.

"Con il superamento della revisione PB – a fronte di una prestazione estremamente deludente – avete concluso il vostro progetto didattico di IS."

— Tullio Vardanega.

"Non posso aiutarvi"

— Alessandro Staffolani.

"Mi avete fatto perdere mesi della mia vita"

— Riccardo Cardin.

Ringraziamenti

Desidero esprimere la mia gratitudine al professor Lamberto Ballan, mio relatore, per l'aiuto e il sostegno che mi ha dato durante la stesura dell'elaborato. Vorrei anche ringraziare, con affetto, i miei genitori per il loro sostegno, il grande aiuto e la loro presenza in ogni momento durante gli anni di studio. Desidero poi ringraziare i miei amici per i bellissimi anni trascorsi insieme e le mille avventure vissute.

Padova, Settembre 2024 Riccardo Zaupa

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage del laureando Riccardo Zaupa presso l'azienda Sanmarco Informatica S.p.A. . Tale periodo, svolto alla conclusione del percorso di studi triennale in Informatica presso l'Università degli Studi di Padova, ha avuto una durata complessiva di trecentoventiore.

Gli obiettivi principali del progetto hanno riguardato l'analisi e l'utilizzo dei servizi AWS per l'addestramento di modelli di Intelligenza Artificiale (AI), finalizzati alla classificazione e all'estrapolazione automatica delle informazioni contenute nelle mail PEC (Poste Elettroniche Certificate). Durante lo stage, è stata eseguita un'analisi dettagliata dei requisiti applicativi e tecnici necessari per implmentare 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 basate su criteri come mittente, destinatario, data e argomento. 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 alla prima occorrenza nel documenti e definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*[g];
- 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 L'azienda

Sanmarco Informatica S.p.A. (logo in figura 1.1) è un'azienda italiana di sviluppo software e consulenza informatica. Da oltre quarant'anni si dedica alla riorganizzazione dei processi aziendali in tutti i settori, progettando e implementando soluzioni digitali integrate.
L'azienda, che ad oggi conta più di 600 dipendenti e oltre 2500 aziende seguite ha come sede principale Villa Ramanelli a Grisignano di Zocco, in provincia di Vicenza, poco distante dai Centri di Ricerca e Sviluppo (CRS) e dal Centro per la Formazione di Vicenza. Conta anche diverse filiati in Trentino-Alto Adige, Friuli-Venezia Giulia, Lombardia, Piemonte, Emilia-Romagna, Toscana, Campania e Puglia.
L'obiettivo principale è l'innovazione e il progresso tecnologico, con l'obiettivo di creare soluzioni software che siano in grado di rispondere alle esigenze dei clienti, garantendo la massima qualità e sicurezza.
L'azienda è organizzata in *Business Unit*, dei centri di competenza specifici e autonomi ma in relazione costante. Ognuna delle quali è specializzata in un settore specifico. La *Business Unit* interessata dallo stage è XC situata nel Centro per la Formazione di Vicenza. Tale team composto da 10 persone, si oocupa di sviluppare e manutenere i servizi di XC.
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 basa su principi di collaborazione, auto-organizzazione e flessibilità. Scrum è un framework Agile che permette di gestire progetti complessi, garantendo la massima trasparenza e la massima flessibilità e suddividendo il progetto in sprint ovvero periodi di tempo relativamente brevi in cui vengono fissati determinati obbiettivi ed attività.
Eventuali ulteriori informazioni sono disponibili sul sito web dell'azienda[1].

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

Figura 1.1: Logo di Sanmarco Informatica

1.2 L'offerta di stage

L'obiettivo dello stage consiste nella catalogazione delle Poste Elettroniche Certificate ([g]PEC), integrando tecnologie di Intelligenza Artificiale ([g]AI) per l'analisi e l'efficienza del processo.
Il modello di apprendimento automatico analizza il contenuto delle PEC e le classifica in base al contenuto.
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. Tale evento mira ad agevolare l'incontro tra studenti e aziende, offrendo la possibilità di svolgere uno stage formativo con specifico riferimento al settore ICT (Information and Communication Technology). Tale settore si riferisce all'insieme delle tecnologie utilizzate per la gestione e la comunicazione delle informazioni, incluse quelle legate all'informatica e alle telecomunicazioni.

1.3 Organizzazione del testo

Il secondo capitolo descrive ...

Il terzo capitolo approfondisce ...

Il quarto capitolo approfondisce ...

Il quinto capitolo approfondisce ...

Il sesto capitolo approfondisce ...

Nel settimo capitolo descrive ...

Capitolo 2

Descrizione dello stage

In questo capitolo verrà descritto il progetto di stage, analizzando il contesto aziendale e le attività svolte durante il periodo di stage.

2.1 Introduzione al progetto

L'elaborazione intelligente dei documenti ([g]IDP) è una tecnologia che automatizza il processo di immissione manuale dei dati da documenti cartacei o immagini digitali, integrandoli con altri processi aziendali digitali. Ad esempio, in un flusso di lavoro aziendale automatizzato, come l'invio di ordini ai fornitori al momento del calo delle scorte, l'IDP può sostituire l'immissione manuale dei dati da parte del team contabile. Invece di inserire manualmente i dati di una fattura ricevuta via e-mail, i sistemi di IDP estraggono automaticamente queste informazioni e le integrano direttamente nel sistema contabile, eliminando ostacoli e riducendo gli errori.

L'IDP offre numerosi vantaggi alle aziende. In termini di [g]Scalabilità, permette di elaborare documenti su larga scala con precisione, evitando errori umani e aumentando l'efficienza operativa. Promuove una cultura dell'efficienza dei costi, automatizzando attività ripetitive e riducendo i costi associati all'elaborazione manuale dei dati. Migliora anche la soddisfazione dei clienti grazie alla gestione più rapida e automatizzata dei documenti, come l'onboarding, le prenotazioni e i pagamenti, consentendo di fornire risposte personalizzate e veloci ai clienti.

Diversi settori traggono beneficio dall'IDP. Nel settore sanitario, facilita la gestione delle cartelle cliniche, migliorando l'estrazione e l'organizzazione dei dati dai documenti medici. Le aziende finanziarie lo utilizzano per automatizzare la gestione delle spese e l'elaborazione delle fatture, semplificando la gestione dei pagamenti. Nel settore legale, l'IDP analizza contratti e documenti legali, utilizzando tecnologie di elaborazione del linguaggio naturale ([g]NLP) per estrarre informazioni chiave. Le aziende della logistica lo impiegano per tracciare spedizioni e documenti di transito, riducendo gli errori umani. Infine, nel settore delle risorse umane, l'IDP semplifica la selezione del personale, gestisce le buste paga e automatizza altre funzioni HR.

Le tecnologie alla base dell'IDP comprendono il riconoscimento ottico dei caratteri ([g]OCR), che converte immagini di testo in dati leggibili dalle macchine, e l'elaborazione del linguaggio naturale (NLP), che analizza e comprende il linguaggio umano. L'automazione robotica dei processi (RPA) consente invece di automatizzare i flussi di lavoro aziendali ripetendo azioni umane predefinite.

Il processo di IDP si articola in diverse fasi: acquisizione e classificazione dei documenti, estrazione dei dati rilevanti tramite OCR e NLP, convalida e successiva elaborazione dei dati nei sistemi aziendali, e apprendimento continuo attraverso algoritmi di machine learning per migliorare l'accuratezza nel tempo. Inoltre, i sistemi di IDP offrono report e analisi per ottimizzare ulteriormente i flussi di lavoro aziendali.

[g]AWS (AWS) supporta l'implementazione dell'IDP attraverso servizi come Amazon Textract, che utilizza il machine learning per estrarre informazioni dai documenti senza interazioni manuali, e Amazon Comprehend, che sfrutta l'NLP per scoprire informazioni preziose nei testi. Entrambi i servizi consentono alle aziende di automatizzare la gestione dei documenti in modo efficiente e sicuro, integrandosi con altre piattaforme aziendali per un flusso di lavoro senza interruzioni.

2.2 Requisiti e obiettivi

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

[Priorità][Id]

- Priorità: indica la priorità dell'obiettivo, può essere obbligatorio o desiderabile;
- Id: composto da due cifre, identifica l'obiettivo in modo univoco rispetto alla priorità.

ID	Categoria	Descrizione
O01	Obbligatorio	Analisi dei servizi AWS per l'addestramento dei modelli AI
O02	Obbligatorio	Addestramento di un modello di apprendimento AI utilizzando i servizi AWS
O03	Obbligatorio	Analisi requisiti applicativi e tecnici per implementare la soluzione richiesta
O04	Obbligatorio	Implementare un modello di apprendimento automatico che analizzi il contenuto delle PEC importate e assegni loro categorie appropriate in base al contenuto (mittente, destinatario, data e argomento)
D01	Desiderabile	Implementare algoritmi di AI in grado di adattarsi e apprendere continuamente dai dati per migliorare le prestazioni del sistema nel tempo. Ciò include l'ottimizzazione dei modelli di apprendimento automatico in base all'esperienza e ai feedback degli utenti

ID	Categoria	Descrizione
D02	Desiderabile	Integrazione con un sistema documentale per l'archiviazione delle PEC creando i metadati necessari con le informazioni estratte e collocandole nella corretta categoria di appartenenza

2.3 Pianificazione

2.3.1 Pianificazione settimanale

Il periodo di stage è stato suddiviso in 8 settimane, durante le quali sono previste le seguenti attività:

Settimana	Dal	Al	Attività
1	24-06-2024	28-06-2024	- Incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste di implementazione - Ricerca, studio e documentazione per inquadramento 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 modello di apprendimento per catalogare le PEC in base al loro contenuto
5	22-07-2024	26-07-2024	- Implementazioni per interfacciarsi con il modello di apprendimento addestrato e per poter catalogare le PEC importate **Milestone:** Completamento obiettivi minimi
6	29-07-2024	02-08-2024	- Implementazione algoritmo di AI per l'autoapprendimento

Settimana	Dal	Al	Attività
7	05-08-2024	09-08-2024	- Studio e documentazione sulle [g]Application Program Interface messe a disposizione dal documentale per poter 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 archiviazione PEC nel documentale **Milestone:** Completamento obiettivi massimi
9	19-08-2024	23-08-2024	- Recupero eventuali ritardi
10	26-08-2024	30-08-2024	- Recupero eventuali ritardi

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

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 Artificial Intelligence (AI) e [g]Machine Learning (ML). 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 (PII, *Personally Identifiable Information*), sentimenti (positivo, negativo, neutro, misto), 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 dataset etichettato per addestrare il modello e supportano le'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 MLOps (*Machine Learning Operations*) 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 (Optical Character Recognition (OCR)) che sfrutta l'apprendimento automatico per identificare e analizzare testo e dati presenti in immagini o documenti. Basato sulla tecnologia di deep learning collaudata e altamente scalabile sviluppata dagli esperti di visione artificiale 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 machine learning 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.

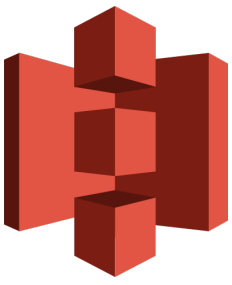


Figura 3.3: Logo di Amazon S3

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 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 machine learning 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 *machine learning* (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 (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 (*foundation models*, 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 modelli di fondazione 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 AI generativa.

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

In aggiunta ai servizi AWS, sono stati utilizzati diversi strumenti di sviluppo per la realizzazione dell'applicazione. Questi strumenti hanno permesso di scrivere, testare e monitorare il codice. Di seguito sono elencati i principali strumenti utilizzati nel corso del progetto.

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, testo, grafici e altri elementi multimediali. Jupyter Notebook supporta diversi linguaggi di programmazione, tra cui Python, R e Julia, e offre un ambiente di sviluppo flessibile e versatile per l'analisi dei dati, la visualizzazione e la prototipazione di modelli di machine learning.

Nel contesto del presente progetto, Jupyter Notebook è stato utilizzato per eseguire analisi preliminari sui dati, testare le funzionalità di Amazon Comprehend e Amazon Textract e sviluppare i modelli di classificazione.

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 Windows, Linux e macOS. Grazie alla sua versatilità e alle numerose estensioni disponibili, Visual Studio Code è stato utilizzato per lo sviluppo del codice dell'applicazione, inclusi i *Lambda functions* e i notebook. Inoltre, l'editor è stato impiegato per redigere e gestire la documentazione del progetto, sfruttando le sue funzionalità avanzate di editing e integrazione con strumenti di controllo di versione.

Figura 3.10: Logo di Visual Studio Code

3.2.3 Git

[g]Git è un sistema di controllo di versione distribuito ampiamente utilizzato per gestire e tracciare le modifiche al codice sorgente durante lo sviluppo software. Nel presente progetto, Git è stato utilizzato per monitorare l'evoluzione del codice sorgente.

Figura 3.11: Logo di Git

3.2.4 Bitbucket

Bitbucket è un servizio di hosting di [g]repository Git basato su cloud. Bitbucket è stato utilizzato per memorizzare il codice sorgente dell'applicazione.

Figura 3.12: Logo di Bitbucket

3.3 Linguaggi di programmazione

Nel corso del progetto sono stati utilizzati diversi linguaggi di programmazione per sviluppare le funzionalità dell'applicazione. Di seguito sono elencati i principali linguaggi utilizzati e le relative caratteristiche.

3.3.1 Python

Python è un linguaggio di programmazione ad alto livello, interpretato, adatto per lo sviluppo di applicazioni web, desktop e mobile. Python è stato utilizzato per la realizzazione delle funzioni Lambda.

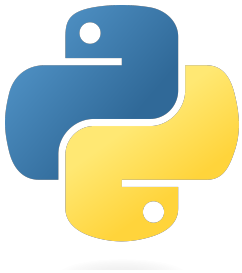


Figura 3.13: Logo di Python

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 Introduzione

Le fasi di un flusso di lavoro per l'Intelligent Document Processing (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.

- **Review and 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. Ove necessario, viene coinvolto un revisore umano per garantire la qualità delle informazioni estratte, 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. In particolare tramite 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.

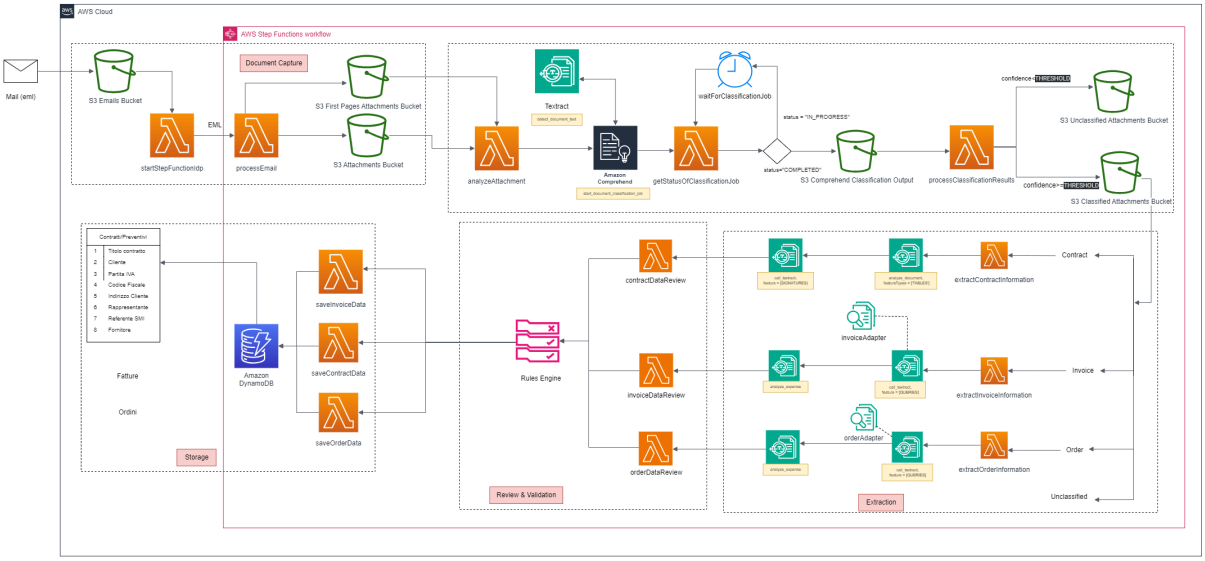


Figura 4.1: Flusso di lavoro per l'Intelligent Document Processing

Di seguito di visualizza la state machine "IdpStateMachine" nel dettaglio.

4.2 Architettura ad alto livello

L'architettura proposta è stata concepita 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.

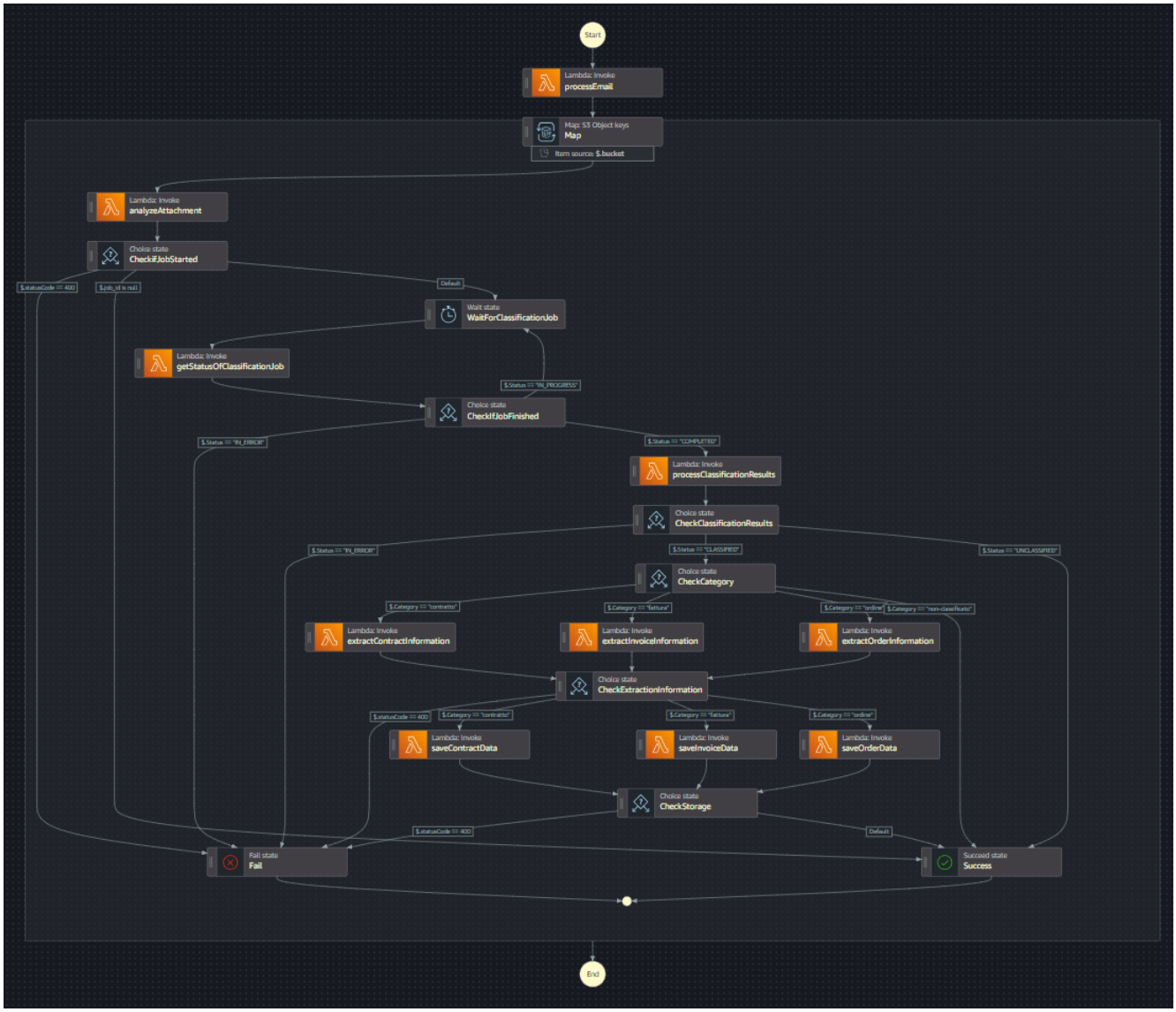


Figura 4.2: State Machine per l'Intelligent Document Processing

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". In aggiunta, le prime pagine dei file PDF vengono salvate nel bucket "S3 First Pages 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`).

Questa sequenza di operazioni permette di strutturare un flusso di lavoro chiaro e diretto, facilitando la successiva classificazione e analisi dei documenti estratti dalle email.

4.4 Classificazione dei documenti

Inizialmente, l'idea era 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 machine learning) si riducono al semplice testo estratto dagli allegati.

La fase di classificazione degli allegati 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. 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, rendendoli disponibili per ulteriori elaborazioni.

- **Processamento dei risultati della classificazione**: Al termine del job di classificazione, la funzione Lambda "processClassificationResults" si attiva per salvare gli allegati classificati nei bucket appropriati. 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 Unclassified Attachments Bucket" e quelli salvati nel bucket "S3 Classified Attachments Bucket" all'interno della cartella "non classificato". Gli allegati presenti nel "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 una classificazione effettuata con un certo grado di sicurezza.

In questo contesto, l'utilizzo di Amazon Textract e Amazon Comprehend si è rivelato cruciale per l'accuratezza della classificazione. Sebbene in una fase iniziale fosse stato considerato l'uso del servizio Amazon Bedrock, 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.

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 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 100 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**: È stata condotta un'analisi approfondita del dataset per valutare la distribuzione delle categorie e garantire un bilanciamento adeguato.
- **Preprocessing**: I dati sono stati preparati per il training, estratti i testi dai documenti PDF e organizzati in un file CSV.
- **Training**: Il modello personalizzato è stato addestrato utilizzando il file CSV creato in precedenza.
- **Valutazione**: Il modello è stato valutato utilizzando metriche standard per valutarne le prestazioni.
- **Test del modello**: Il modello è stato testato utilizzando nuovi documenti per verificare la sua capacità di generalizzazione.

4.4.1.1 Analisi del dataset

L'analisi del dataset è fondamentale per comprendere la distribuzione delle categorie e valutare l'adeguatezza dei dati per il training. 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.

4.4.1.2 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.
- 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.

4.4.1.3 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. Questo processo è stato gestito tramite il servizio Custom Classifier, che ha permesso di creare un modello ottimizzato per le esigenze specifiche del progetto.

4.4.1.4 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.

4.4.1.5 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 capacità del modello.

4.4.1.6 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. Fondamentalmente sono stati analizzati diversi metodi utilizzando differenti servizi per aderire a tale scopo:

- Comprehend custom entities
- Amazon Bedrock
- features di textract

Digressione sui vantaggi e svantaggi ... Alla fine si è optato per le seguenti opzioni:

- Custom queries per le fatture
- Custom queries per gli ordini
- Analisi delle tabelle e dei form per i contratti

C'è da sottolineare che per ogni informazione estratta viene anche riportata la percentuale di confidenza. Il flusso per ogni categoria è il seguente:

- Quando un file viene caricato nel bucket relativo ai documenti classificati tale azione scatena l'esecuzione di una lambda apposita per il tipo di documento
- Al termine dell'esecuzione tali informazioni estratte vengono passate alla fase successiva

4.5.1 Estrazioni delle informazioni dai contratti

Per tale fase essendo i contratti della stessa forma, (una tabella con le seguenti informazioni ...) si è optato per un'opzione poco costosa ma comunque efficiace. Tale soluzione consiste nell'identificare tale tabella ed estrarne i campi in base alla conoscenze note.

4.5.2 Estrazione delle informazioni dalle fatture e degli ordini

Per tale fase si è pensato all'utilizzo di custom queries (adapter) di textract dato che tali documenti possiedono una struttura variabile. L'utilizzo di analisi delle fatture tramite la funzione apposita di textract è stata considerata ma poi scartata. Per gli ordini invece si è pensato di utilizzarla per ricavarne gli articoli in modo piu' diretto e sicuro.

4.6 Persistenza dei dati

In questa fase l'obiettivo è far persistere i dati. La scelta è ricaduta su Amazon DynamoDB. Il flusso è il seguente:

- Per ogni categoria (contratti, ordini, fatture) è creata una lambda, tale lambda salva i risultati delle informazioni estratte in DynamoDB nelle tabelle Ordini, Contratti, Fattura, Articoli_Fatture, Articoli_Ordini

4.7 Analisi dei costi

Capitolo 5

Sviluppi futuri

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.
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'estrazioni delle informazioni associate a ciascuna categorie.

5.3 Completamento delle informazioni

Si possono completare le informazioni mancanti non estratte interrogando il database DynamoDB. Questo lavoro si può fare tra lo step di 2 e lo step 3.

5.4 Sviluppo di un'interfaccia grafica

Capitolo 6

Conclusioni

Lorem [g]SDK

Lorem Application Program Interface

6.1 Consuntivo finale

Ipsum

6.2 Raggiungimento degli obiettivi

Sit amet

6.3 Conoscenze acquisite

6.4 Valutazione personale

Acronimi e abbreviazioni

AI Artificial Intelligence. i, 7, 29

API Application Programming Interface. i, 29

AWS Amazon Web Services. i, 7, 29

IDP Intelligence Document Processing. i, 29

ML Machine Learning. i, 7, 29

NLP Natural Language Processing. i, 29

OCR Optical Character Recognition. i, 8, 29

PEC Posta Elettronica Certificata. i, 29

SDK Software Development Kit. i, 30

UML Unified Modeling Language. i, 30

Glossario

Active learning Nell'ambito del machine learning per active learning si intende .... i, 24, 29

Agile Nell'ambito dell'ingegneria del software con il termine Agile si intende .... i, 1, 29

AI Per aritificial Intelligence (AI) si intende .... i, 2, 4, 5, 28

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, 6, 27, 28

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, 4, 5, 28

Bias Nell'ambito del machine learning per bias si intende .... i, 21, 29

Bucket Nel contesto di AWS, per bucket si intende .... i, 29

Git Git è un sistema di controllo di versione distribuito gratuito e open source progettato per gestire tutto, dai piccoli ai grandi progetti, con velocità ed efficienza. i, 15, 29

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, 3, 4, 28

ML Per Machine Learning (ML) si intende .... i, 20, 28

NLP Natural Language Processing (NLP) è ... i, 3, 4, 7, 28

OCR Optical Character Recognition (OCR) è .... i, 4, 28

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, 2, 4–6, 28

Repository Con il termine repository si intende .. . i, 15, 30

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, 3, 30

Scrum In ingegneria del software, per Scrum si intende .... i, 1, 30

SDK A software development kit (SDK) is a collection of software development tools in one installable package. They facilitate the creation of applications by having a compiler, debugger and sometimes a software framework. They are normally specific to a hardware platform and operating system combination. To create applications with advanced functionalities such as advertisements, push notifications, etc; most application software developers use specific software development kits. i, 27, 28

Serverless Per serverless si intende .... i, 7, 11, 12, 30

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, 28

Bibliografia

Books

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

Articles

Einstein, Albert, Boris Podolsky e Nathan Rosen. «Can Quantum-Mechanical Description of Physical Reality be Considered Complete?» In: *Physical Review* 47.10 (1935), pp. 777–780. DOI: 10.1103/PhysRev.47.777.

Siti web consultati

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

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

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

aws samples. URL: https://github.com/aws-samples/aws-ai-intelligent-document-processing.

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

comprehend samples. URL: https://github.com/aws-samples/amazon-comprehend-examples/blob/master/building-custom-classifier/BuildingCustomClassifier.ipynb.

flywheel. URL: https://aws.amazon.com/it/blogs/machine-learning/introducing-the-amazon-comprehend-flywheel-for-mlops/.

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

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

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

sdk samples. URL: https://github.com/awsdocs/aws-doc-sdk-examples.

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

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