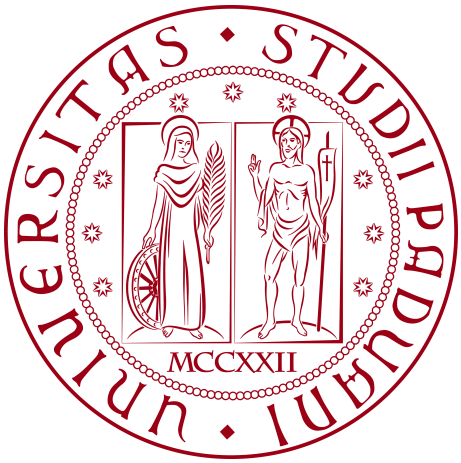
Università degli Studi di Padova

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.

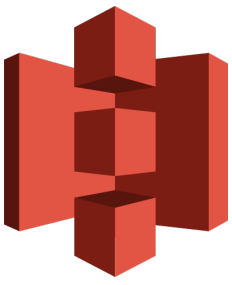


Figura 3.3: Logo di Amazon S3

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.

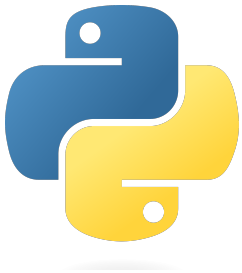


Figura 3.13: Logo di Python

Capitolo 4

Progettazione e codifica

Breve introduzione al capitolo

4.1 Introduzione

Partendo da quella che è la richiesta dell'azienda ospitante (catalogazione ed elaborazione delle mail e dei documenti allegati), ho individuato diverse fasi per il processo di elaborazione dei documenti dalle email:

- Estrazione degli allegati presenti in un'email
- Classificazione dei documenti
- Estrazione delle informazioni importanti dai documenti
- Revisione e valutazione umana delle informazioni estratte
- Persistenza dei dati

4.2 Estrazione degli allegati

In questa fase ,le indicazioni iniziali dell'azienda consistevano nell'analizzare il contenuto delle eventuali email ,per poi effettuarne una classificazione basata sull'elaborazione del linguaggio naturale e dei metadati contenuti. Tuttavia, con il chiarimnento delle categorie prese in analisi durante lo stage (ordini, fatture e contratti) , ho scelto di concentrarmi sull'estrazione degli allegati presenti nelle stessa piuttosto che sul contenuto della email. Tale visione è supportata dal fatto che i documenti di interesse per l'azienda sono spesso allegati delle mail, quindi tale estrazione determina un flusso di lavoro più chiaro e diretto, oltre al fatto che spesso il contenuto della mail non è rilevante o lo è solo in parte. Dunque, in questa fase il file di input è un file .eml , mentre l'output sono gli allegati presenti nella mail. Per progettare tutto ciò, ho pensato ai seguenti servizi:

- Un [g]bucket contenente i file .eml da analizzare

- Un bucket contente gli allegati estratti dalle mail
- Una funzione lambda che viene attivata dall'inserimento di un file .eml all'interno del bucket di input e che elabora tale file per ottenere degli allegati di output

4.3 Classificazione dei documenti

Dovendo classificare le email in base al loro contenuto, ho analizzato diverse strade possibili , come quella di utilizzare modelli di ML proposti da Sagemaker per classificare le email. Tuttavia, con il chiarimento delle categorie prese in analisi durante lo stage (ordini, fatture e contratti) , ho scelto di concentrarmi sull'estrazione degli allegati presenti nelle stessa piuttosto che sul contenuto della email. Questo cambio di prospettiva ha portato a una semplificazione del processo di classificazione, in quanto i metadati (denominati anche features nel campo del machine learning) si riducono al semplice testo estratto dagli allegati. Dunque, per poter classificare un documento in una delle categorie di interesse, ho pensato di utilizzare un modello di machine learning che prendesse in input il testo estratto e restituisse la categoria di appartenenza. In questo senso ,l'utilizzo di strumenti come Textract e Comprehend di AWS si è rivelato molto utile, in quanto permette di estrarre il testo dai documenti e di analizzarlo per ottenere informazioni utili. Tuttavia, c'è anche da sottolineare come in una fase iniziale si sia dibattuto riguardo l'utilizzo al loro posto del servizio Bedrock con claude-3. Tale servizio è stato poi scartato a favore di un modello customizzato e maggiormente adatto alla soluzione. In questa fase è necessario distinguire due tipi di flusso di lavoro:

- Flusso di lavoro per il training del modello
- Flusso di lavoro per la classificazione

4.3.1 Flusso di lavoro per il training del modello

In questa fase per poter addestrare un modello di Comprehend è necessario disporre di un dataset ampio ,significativo e bilanciato per poter distingure le categorie di interesse. Inoltre, è necessario disporre di un dataset etichettato, in cui ogni documento è associato alla sua categoria di appartenenza. Durante la fase di etichettatatura, sono emerse delle considerazioni importanti. Inanzitutto, i file da analizzare sono per lo piu file pdf (di documenti scansionati o meno), per tale motivo per la fase di training sono stati utilizzati unicamente file con tale estensione. Inoltre, si è scelto, in accordo con il tutor aziendale ,di utilizzare per il training unicamente le prime pagine di tali documenti per diversi motivi : spesso le prime pagine contengono le informazioni più importanti e rilevanti per la classificazione, inoltre, il costo di analizzare un documento è proporzionale al numero di pagine, quindi riducendo il numero di pagine si riducono i costi, assumendo anche il fatto che rispetto ai documenti incontrati il numero di pagine variava fino a 100 pagine. Queste due scelte (utilizzo di file pdf e utilizzo delle prime pagine) hanno portato a una riduzione della varietà dei dati e quindi a una riduzione della capacità del modello di generalizzare su nuovi dati ,creando dei potenziali [g]bias nel modello. Di seguito vengono riportati i dati e il loro numero per trainare la prima versione del modello

- 100 documenti per la categoria ordini
- 100 documenti per la categoria fatture
- 100 documenti per la categoria contratti
- 100 documenti per la categoria non classificato

Per il processo chiamato [g]active learning è stato scelto l'utilizzo di un servizio incluso in Comprehend introdotto recentemente da aws chiamato flywheel. Tale processo segue i seguenti passi:

- Viene creato un dataset flywheel
- Viene inizializzata un'iterazione flywheel
- In base ai risultati dell'iterazione viene scelto se attivare il nuovo modello formatosi in base a parametri scelti in precedenza

4.3.1.1 Analisi del dataset

Percentuale di oridine, fattura, contratti e non classificato

4.3.1.2 Preprocessing

- Estrazione del testo tramite Amazon Textract
- Creazione del file csv
- Caricamento del file di training csv trmite flywheel

4.3.1.3 Training

- Creazione di una versione del classificatore su Custom Classifier

4.3.1.4 Valutazione

4.3.1.5 Test del modello

4.3.2 Flusso di lavoro per la classificazione

Per tale fase, ho scelto di utilizzare i seguenti servizi:

- una funzione lambda che riceve in input un'allegato di qualsiasi tipo e (in base se è un pdf o meno) fa partire il processo di classificazione mediate il modello attivo di comprehend
- un modello attivo di comprehend che utilizza le funzionalità di textract per estrarre il testo in chiaro dal pdf ricevuto in input e restituisce la categoria di appartenenza con un certa confidenza
- tramite un'ulteriore funzione lambda viene analizzata la confidenza e in base a questo viene scelto se salvare l'allegato in un bucket che contiene gli allegati non classificati oppure in un bucket con alto grado di confidenza

4.4 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

4.4.1 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.

4.4.2 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.

4.5 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

4.6 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, 25

API Application Programming Interface. i, 25

AWS Amazon Web Services. i, 7, 25

IDP Intelligence Document Processing. i, 25

ML Machine Learning. i, 7, 25

NLP Natural Language Processing. i, 25

OCR Optical Character Recognition. i, 8, 25

PEC Posta Elettronica Certificata. i, 25

SDK Software Development Kit. i, 26

UML Unified Modeling Language. i, 26

Glossario

Active learning Nell'ambito del machine learning per active learning si intende .... i, 19, 25

Agile Nell'ambito dell'ingegneria del software con il termine Agile si intende .... i, 1, 25

AI Per aritificial Intelligence (AI) si intende .... i, 2, 4, 5, 24

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, 23, 24

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, 18, 24

Bias Nell'ambito del machine learning per bias si intende .... i, 18, 25

Bucket Nel contesto di AWS, per bucket si intende .... i, 17–19, 25

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, 25

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, 24

ML Per Machine Learning (ML) si intende .... i, 18, 24

NLP Natural Language Processing (NLP) è ... i, 3, 4, 7, 24

OCR Optical Character Recognition (OCR) è .... i, 4, 24

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, 24

Repository Con il termine repository si intende .. . i, 15, 26

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, 26

Scrum In ingegneria del software, per Scrum si intende .... i, 1, 26

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, 23, 24

Serverless Per serverless si intende .... i, 7, 11, 12, 26

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, 24

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/.