Università degli Studi di Milano
Facoltà di Scienze e Tecnologie
Corso di Laurea in Informatica

CLASSIFICAZIONE DI FATTURE ELETTRONICHE PASSIVE ATTRAVERSO ALGORITMI DI MACHINE LEARNING

Relatore: Prof.ssa Silvana CASTANO

Tesi di:
Alessandro Fabio FARÈ
Matricola: 910928

Anno Accademico 2021-2022

dedicato alla mia famiglia

Ringraziamenti

Dedico questo spazio a tutte le persone che mi hanno permesso di arrivare fin qui e di portare a termine questo lavoro di tesi.

Un infinito ringraziamento alla mia famiglia che mi ha sempre sostenuto durante tutto il mio percorso universitario, perchè senza di loro non avrei mai avuto la possibilità di arrivare fin qui. Un pensiero va anche ai miei nonni che mi hanno protetto.

Un sentito grazie alla mia relatrice, Prof.ssa Silvana Castano, per la sua infinita disponibilità e tempestività ad ogni mia richiesta.

Ringrazio inoltre l'azienda Digital Technologies in cui ho svolto il tirocinio e sviluppato il progetto, in particolar modo il mio tutor aziendale Massimo Serranò e il responsabile IT Emanuele Crespi, per l'ospitalità ricevuta e per avermi dato l'opportunità di mettermi in gioco e fare un'esperienza che sarà preziosa per il mio futuro.

Non posso poi fare a meno di ringraziare tutti i miei amici che mi hanno sempre sostenuto e supportato in questi anni, e i miei compagni di corso, Alberto, Dario, Davide, Manuel, Marco e Vladislav, con cui ho condiviso il corso di laurea e che, anche durante le fatiche e lo sconforto che hanno caratterizzato il mio percorso di studi, mi sono sempre stati accanto e aiutato fino al raggiungimento del mio traguardo.

Indice

Ringraziamenti **iii**

1 Introduzione **1**
1.1 Contesto applicativo . . . 3

2 Classificazione di documenti e Machine Learning **4**
2.1 Classificazione del testo . . . 4
2.2 Estrazione dei dati . . . 7
2.3 Preprocessing e pulizia del testo . . . 7
2.3.1 Selezione dei dati . . . 8
2.3.2 Stemming . . . 8
2.4 Indicizzazione dei documenti . . . 9
2.4.1 Riduzione dimensionale . . . 9
2.5 Machine Learning . . . 10
2.5.1 Algoritmi . . . 12

3 Progettazione e implementazione del sistema **14**
3.1 Definizione del problema e analisi dei requisiti . . . 14
3.1.1 Analisi delle fatture . . . 16
3.1.2 Analisi delle fatture estratte . . . 20
3.1.3 Analisi Machine Learning . . . 21
3.2 Raccolta dei documenti . . . 22
3.3 Estrazione dei dati . . . 23
3.4 Preprocessing . . . 23
3.4.1 Data di emissione . . . 24
3.4.2 Numero fattura . . . 24
3.4.3 Importi e IVA . . . 24
3.4.4 DDT . . . 25
3.4.5 Valuta . . . 25
3.5 Indicizzazione dei documenti . . . 26
3.6 Training e Testing . . . 27

3.7 Costruzione dei modelli 27
3.8 Esecuzione degli algoritmi 29

4 Analisi dei risultati **31**
4.1 Validità dei risultati 31
4.2 Risultati di classificazione 33
4.2.1 K-nearest neighbors 34
4.2.2 Random Tree 34
4.2.3 Random Committee 35
4.2.4 Risultati Migliori 36
4.3 Considerazioni 37
4.3.1 Metriche di accuratezza 38

5 Conclusioni e sviluppi futuri **40**
5.1 Conclusioni 40
5.2 Sviluppi futuri 42

Capitolo 1

Introduzione

Negli ultimi anni si è assistito a un'esplosione nella produzione di documenti in formato digitale. Grazie alla maggiore disponibilità di strumenti hardware e software per la generazione di dati digitali e al protocollo http (HyperText Transfer Protocol) definito da Berners-Lee nel 1991, si è reso possibile far consultare a milioni di persone una grandissima quantità di documenti.

Una buona parte dei documenti presenti nel World Wide Web è sottoforma di testo libero scritto in linguaggio naturale non classificato, nè strutturato, per il quale sta nascendo una sempre più crescente necessità di soluzioni per la sua archiviazione e organizzazione al fine di recuperare informazioni importanti.

Gestire i documenti non strutturati rappresenta il principale trend con cui devono fare i conti tutte le organizzazioni che si trovano oggi ad affrontare le criticità legate a un data management sempre più complesso. I dati contenuti nei documenti sono tutti quei metadati senza un modello predefinito che non appartengono a un database ma che forniscono un tipo di informazione qualitativa. Le organizzazioni che vogliono sfruttare l'immenso patrimonio informativo incorporato in questi dati si stanno orientando su nuove soluzioni tecnologiche volte a migliorare la raccolta e l'organizzazione dei dati. L'impresa ha bisogno oggi di strumenti che siano in grado di indicizzare, interpretare ed estrarre valore dai documenti destrutturati.

Un approccio a riguardo, che sta prendendo sempre più piede negli ultimi anni, è l'utilizzo di algoritmi di Machine Learning per la realizzazione di sistemi per la classificazione automatica dei testi contenuti nei documenti digitali. Il Machine Learning permette di classificare testi digitali espressi in linguaggio naturale etichettando in maniera automatica collezioni di documenti associandoli a categorie definite a priori, imparando da un set di documenti preclassificato. L'obiettivo di questi sistemi è aumentare la rilevabilità delle informazioni e rendere disponibile o utilizzabile tutta la conoscenza acquisita per supportare il processo decisionale strategico delle organizzazioni.

Lo scopo del progetto qui descritto consiste nella realizzazione di un sistema per la classificazione automatica dei dati di fatture elettroniche passive attraverso algoritmi di Machine Learning. Le fatture elettroniche passive contengono importanti informazioni per le aziende, ed è diventato ormai fondamentale gestire questi dati e immagazzinarli. Allo scopo di minimizzare l'attività amministrativa di gestione delle fatture, è desiderabile un applicativo semplice e veloce per il recupero, la gestione e archiviazione di quest'ultime.

A questo scopo, sono state raccolte fatture a livello globale, provenienti da diverse parti del mondo e contenenti l'informazione necessaria per poter realizzare un sistema idoneo che possa gestire fatture di un qualsiasi fornitore. I formati dei dati e i layout delle fatture possono differire da un'azienda a un'altra, il che rende ancor più complicata la loro categorizzazione. Alcune fatture potrebbero inoltre aver avuto problemi di stampa peggiorando così l'estrazione dei dati e la successiva classificazione. Al fine di semplificare l'etichettatura dei dati delle fatture, la classificazione automatica del testo attraverso l'apprendimento automatico può essere un'ottima soluzione.

Il progetto di tesi consiste in un applicativo interamente realizzato in linguaggio Java che si articola in tre sottosistemi:

1. Estrazione e preprocessing del testo: rilevazione del testo contenuto nelle fatture elettroniche passive attraverso un sistema di riconoscimento ottico dei caratteri, OCR, ed elaborazione e selezione dei dati di interesse in base alle categorie con cui vogliamo classificare i documenti digitali.

2. Indicizzazione dei documenti: trasformazione dei dati estratti non strutturati in una rappresentazione compatta del suo contenuto, creando dei dataset con una struttura matriciale contenente informazioni fondamentali relative ai dati per poter allenare i modelli di Machine Learning.

3. Machine Learning: costruzione dei modelli di Machine Learning ed esecuzione degli algoritmi, con successiva analisi dei risultati.

Il sistema permette di fornire uno strumento automatizzato per la gestione e archiviazione dei dati contenuti nelle fatture. È stato anche pensato per poter precompilare i campi delle piattaforme di gestione delle fatture delle aziende che, a fronte dell'arrivo di una nuova fattura, si trovano a dover identificare all'interno di essa i campi necessari per la sua gestione e archiviazione nei database. Grazie alla classificazione automatica del testo lasceremmo all'amministratore del sistema l'unico compito di ispezionare i campi della fattura per verificarne l'accuratezza e in seguito registrare la fattura, risparmiando molto tempo. Anche in uno scenario in cui la maggior parte dei campi vengano precompilati correttamente si avrebbe un impatto positivo sul tempo complessivo speso per ciascuna fattura e rappresenterebbe quindi un miglioramento del sistema attuale.

1.1 Contesto applicativo

Il progetto è stato sviluppato presso Digital Technologies, una società di consulenza con sede a Milano che nasce con l'intento di sviluppare soluzioni innovative per il miglioramento dei processi aziendali di grandi gruppi industriali.

Una delle soluzioni innovative implementata da Digital Technologies riguarda la realizzazione di una piattaforma per la gestione delle fatture attive e passive. Per il ciclo passivo delle fatture la soluzione consente di ricevere, visualizzare e consultare le fatture passive ricevute e transitate dallo SdI. È evidente che una categorizzazione automatica dei dati di tali fatture può portare a un grosso risparmio di tempo.

L'attuale sistema utilizzato in Digital Technologies non utilizza la classificazione automatica del testo, ma si serve di funzioni per categorizzare i dati grazie alla semantica e a dati precedentemente estratti, senza l'utilizzo di algoritmi di Machine Learning. Sono stati implementati anche template specifici per determinati fornitori che posizionano certi dati importanti delle fatture in posizioni non convenzionali, complicando l'estrazione e la categorizzazione degli stessi. Con un sistema tale, quando occorre gestire fatture con layout particolari, chi amministra il sistema deve trovare ulteriori regole per l'estrazione dei dati, andando a complicare le logiche dietro all'applicativo.

Quando invece vengono utilizzati gli algoritmi di Machine Learning, l'unico compito che spetta all'amministratore è quello di impartirgli delle regole generali, e , a fronte di cambiamenti inconsueti dei layout delle fatture, il sistema si adegua cambiando il suo flusso di categorizzazione in base ai nuovi dati.

A fronte di ciò, l'azienda sta attuando un'evoluzione del sistema precedentemente utilizzato, introducendo la tecnologia RPA, Robotic Process Automation, da affiancare al Machine Learning, creando una sinergia che possa automatizzare l'attività di gestione delle fatture. Una RPA arricchita di algoritmi di Machine Learning permette di garantire performance migliori e anche di gestire qualche eccezione. Consente all'automazione di migliorare col tempo, sia in termini di performance nel riconoscimento testuale che in termini di flessibilità rispetto a documenti con campi invertiti o posizionati in modo non del tutto allineato alla norma.

Capitolo 2

Classificazione di documenti e Machine Learning

2.1 Classificazione del testo

Il termine classificazione del testo viene usato per indicare task diversi. In questa tesi verrà usato come termine per indicare l'assegnamento di un documento in una categoria presa da un insieme predefinito. Si può definire formalmente nel seguente modo [8]:

> Dato un set iniziale di documenti $\mathcal{D} = \{d_1, d_2,...,d_{|D|}\}$,
> pre-classificati in $\mathcal{C} = \{c_1,c_2,...,c_{|C|}\}$ categorie, si dovrebbe ottenere una funzione ϕ: $\mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$, la quale indica se un documento d $\in \mathcal{D}$ deve appartenere alla categoria c $\in \mathcal{C}$ (ϕ (d, c) = T) o non deve appartenere (ϕ (d, c) = F). Tale funzione deve essere il più simile possibile alla funzione "target" ϕ' : $\mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$, la quale è considerata come la giusta funzione classificatrice.

Si assume che le categorie siano etichette simboliche e la classificazione avvenga solo attraverso il contenuto del documento e non per conoscenza esterna.

Dato che la classificazione del testo si basa sulla semantica, e visto che è una nozione del tutto soggettiva, ne segue che l'appartenenza di un documento a una categoria non può essere decisa deterministicamente. Questo è spiegato dall' Inter-indexer inconsistency: quando due esperti umani decidono se classificare un documento d_j sotto c_i, si possono trovare in disaccordo.

Sebbene la classificazione del testo possa essere fatta manualmente e in maniera dettagliata e accurata, essa richiede molto tempo. Un approccio automatico è invece più veloce, scalabile e oggettivo e fornisce alle organizzazioni una classificazione più sistematica e coerente.

Ci sono stati negli anni diversi approcci per la creazione di un classificatore di testi automatico.

Molto popolare negli anni '80 è il cosiddetto Knowledge-base Engineering (KBE), utilizzato per raccogliere, organizzare e riutilizzare le conoscenze tecniche necessarie per automatizzare consistenti porzioni del processo industriale. L'idea centrale della KBE è quella di combinare le informazioni, provenienti da una sorprendente varietà di fonti, per arrivare alla definizione di strategie produttive ottimizzate e frutto di automatismi e relazioni. Funziona recependo come input tutti i dati necessari per lo sviluppo di una determinata mansione; applica successivamente regole per arrivare al progetto di un sistema senza o quasi il controllo passo per passo di una figura umana.

Tali regole sono del tipo:

> if <DNF Formula> then $< c_i >$ else $< \neg c_i >$, dove DNF è una Forma Normale Disgiuntiva e c_i è una categoria i-esima.

Le regole venivano definite da un knowledge engineer con l'aiuto di un esperto di dominio.

L'esempio più famoso è il sistema Construe progettato dal Carnegie Group per l'agenzia di stampa Reuters. Il sistema da loro costruito, chiamato TIS (Topic Identification System), è un sistema di categorizzazione che assegna termini di indicizzazione a notizie di stampa in base al loro contenuto utilizzando l'approccio KBE. TIS ha sostituito l'indicizzazione umana utilizzata precedentemente, la quale si basava su un servizio di informazioni online comprendente notizie indicizzate per paese e tipo di notizia. L'indicizzazione TIS è paragonabile all'indicizzazione umana nella precisione complessiva ma costa molto meno, è più consistente ed è disponibile molto più rapidamente.

Come si può intuire, l'approccio KBE è un primo esempio di classificazione automatica, che tende a sostituire la classificazione manuale in termini di performance e oggettività, soprattutto in un mondo come quello di oggi dove i contenuti digitali sono sempre di più. Ciò nonostante, KBE presenta alcuni svantaggi:

- Se si deve modificare l'insieme di categorie, è necessario nuovamente l'aiuto dell'esperto di dominio.
- Se si vuole cambiare dominio del classificatore occorre chiamare un nuovo esperto di dominio e riniziare il lavoro.

A partire dagli anno '90 ha cominciato a svilupparsi l'approccio Machine Learning. Nel Machine Learning un processo induttivo costruisce automaticamente un classificatore per una determinata categoria c_i, osservando le caratteristiche, note anche con il rispettivo termine inglese features, di un insieme di documenti che sono stati precedentemente classificati sotto c_i o $\neg c_i$ da un esperto di dominio. Non si

costruisce quindi un classificatore, ma un costruttore di classificatori che va bene per ogni dominio.

Al task di classificazione testuale si possono aggiungere dei vincoli: per un dato k, esattamente k elementi del dominio $\mathcal{C}$ vengono assegnati a un documento d_j.

In base a questi vincoli possiamo raccogliere i task in tre diverse tipologie:

- Single Label: solo una categoria può essere assegnata a un documento (k=1).
- Multi Label: zero o più categorie possono essere assegnate a un documento.
- Binary: un documento o appartiene a c_i o appartiene a $\neg c_i$.
- Binary Multi Label: un algoritmo di binary classification può essere usato anche per multilabel: si trasforma il problema di classificare sotto le categorie $\{c_1...c_n\}$ in $\mathcal{N}$ problemi indipendenti di classificazione binaria sotto le categorie $\{c_i, \neg c_i\}$.

Un'altra incidenza da considerare nella classificazione del testo sono le classificazioni Document Pivoted e Category Pivoted, due approcci differenti di classificare:

- Document Pivoted: dato un documento vogliamo trovare tutte le categorie sotto il quale può essere classificato.
- Category Pivoted: data una categoria vogliamo trovare tutti i documenti che possono essere classificati sotto di essa.

L'approccio Document Pivoted è più adatto quando la classificazione viene ad esempio usata per filtrare le email, cioè quando i documenti sono accessibili in tempi diversi. La classificazione Category Pivoted è invece utilizzata quando una nuova categoria dev'essere aggiunta al set corrente di categorie, dopo che i documenti sono già stati classificati e necessitano quindi di una riclassificazione con la nuova categoria.

La classificazione può essere infine definita hard o ranking, a seconda del valore restituito dal classificatore. Hard quando il classificatore restituisce un valore booleano e Ranking quando restituisce un valore binario $[0, 1]$.

Nel progetto è stata utilizzata una classificazione di tipo Single Label, con un approccio Category Pivoted e con un valore restituito booleano, quindi di tipo Hard.

Sono stati classificati i dati associandoli a otto campi diversi delle fatture utilizzate come categorie, ciascuno con formati differenti in base al fornitore. La categorizzazione di questi campi risulta complicata in virtù del fatto che il layout di una fattura di un certo fornitore può risultare molto diversa rispetto a una di un altro fornitore; non cambiano solo i formati ma anche le coordinate cartesiane dei campi delle fatture all'interno del documento, informazione molto importante per gli algoritmi di Machine Learning.

La sfida maggiore della classificazione del testo è quella di trovare delle regole generali di estrazione e indicizzazione in modo da creare dei dataset da dare poi in pasto agli algoritmi di Machine Learning che possano essere sufficienti a far comprendere a quest'ultimi come analizzare i dati e attuare il processo di categorizzazione.

2.2 Estrazione dei dati

Una prima fase importante nel processo di classificazione del testo riguarda l'estrazione dei dati dai documenti. I documenti possono essere di diversi formati, PDF o immagini, e non sempre è facile estrarne tutti i dati che contengono.

Per l'estrazione dei dati è stata utilizzata la tecnologia OCR, dall'inglese Optical Character Recognition. I sistemi di riconoscimento ottico dei caratteri sono programmi dedicati al rilevamento dei caratteri contenuti in un documento e al loro trasferimento in testo digitale leggibile da una macchina. In particolare come sistema OCR è stato usato OCR Tesseract, un progetto open source nato nel 1985 che ancora oggi è aggiornato ed è stato supportato anche da Google. Supporta diversi formati di output, può riconoscere più di cento lingue ed è uno dei sistemi OCR disponibili più accurati, assieme a OCR Opus [3].

Una volta terminata la fase di estrazione dei dati, è necessario trasformare i documenti testuali dal formato destrutturalizzato in cui si trovano ad un formato strutturato adatto per poter applicare gli algoritmi di Machine Learning. Per completare questa conversione bisogna applicare correttamente tutte le fasi elencate nei prossimi paragrafi, altrimenti si possono ottenere risultati insoddisfacenti.

2.3 Preprocessing e pulizia del testo

L'attività di trasformazione e preprocessing dei dati è molto importante in qualsiasi task di Machine Learning, ma nel contesto della classificazione testuale e più in generale dell'elaborazione del linguaggio naturale è molto rilevante.

Un contenuto testuale generalmente contiene molta informazione al suo interno, e non tutta quest'informazione è importante ai fini della classificazione da svolgere. Gran parte del contenuto testuale molto spesso non ha alcuna rilevanza nel risultato prodotto; in alcune occasioni può rappresentare rumore che va a distorcere la classificazione prodotta dal modello.

Nel linguaggio naturale un concetto può essere espresso con termini che hanno stessa radice ma declinazione differente. Quei termini rappresentano lo stesso concetto, ma , avendo una declinazione differente, il modello di Machine Learning potrebbe fare difficoltà a comprendere che si tratti di un concetto similare a quanto già visto. Anche la punteggiatura molto spesso introduce solo rumore nell'analisi del testo.

Per tutti questi motivi è molto importante svolgere delle attività di preprocessing del testo in modo da minimizzare il più possibile tutti gli effetti non desiderati causati da informazione non rilevante presente nel testo.

2.3.1 Selezione dei dati

Una fase importante nella fase di preprocessing è la selezione dei dati. Se questa fase venisse eseguita in modo approssimativo e non relativo alla descrizione del problema, si rischierebbe di ottenere risultati non esatti. Occorre manipolare i dati estraendo un insieme di elementi rilevanti che serviranno poi per gli algoritmi di Machine Learning.

Quando si gestiscono diversi tipi di documenti digitali, un'altra difficoltà è la presenza di errori, come errori di ortografia o errori grammaticali di vario tipo. La gestione di questi errori in una certa misura è fondamentale per qualsiasi tipo di classificazione del testo.

È buona norma anche fare uso di una lista di stop-words, che tradotto significa letteralmente "parole da fermare". Questa lista contiene un elenco di termini che non devono essere considerati perchè non sono rilevanti per i fini che s'intendono realizzare. Al suo interno presenta infatti numeri, caratteri speciali e tutti quei vocaboli che, dopo una scrupolosa analisi delle frequenze, risultano essere comuni. I benefici di tale lista sono la riduzione delle quantità di parole, abbassando così le risorse computazionali del processo, e l'aumento della qualità dei dati.

Conviene infine convertire i termini alla forma minuscola applicando così la funzionalità detta casefolding.

2.3.2 Stemming

Si definisce stemming, il processo di riduzione della forma flessa (i.e., una qualsiasi variazione morfologica) delle parole, alla forma base detta radice o meglio "tema".

Il termine stemming deriva dall'inglese stem, ossia stelo. Si considera la radice di una parola come uno stelo da cui si diramano tutte le parole varianti, come un albero, che compone la famiglia delle parole.

Il processo di stemming consiste nel sostituire in un documento tutte le parole con le relativi radici. Il risultato finale è una versione del testo con la stessa quantità di termini ma con meno varianti.

Questo procedimento non è obbligatorio per gli obiettivi di classificazione dei documenti ma, in alcuni casi, può offrire un piccolo beneficio. Si può infatti affermare che un effetto favorevole è la diminuzione del numero di termini distinti nel testo e l'incremento di frequenza delle occorrenze di alcuni termini. Un altro aspetto positivo da considerare è che l'unione di termini simili nella stessa radice potrebbe garantire

risultati migliori in fase di classificazione, soprattutto per quanto riguarda la fase di indicizzazione dei documenti, necessaria per gli algoritmi di Machine Learning.

La creazione di un algoritmo di stemming è estremamente dipendente dal contesto in cui si sta lavorando e soprattutto dalla lingua utilizzata. Nel progetto di tesi sono state utilizzate regole generali per la normalizzazione dei termini nelle fatture, ciascuna differente in base ai campi considerati per la classificazione, applicando processi di stemming che andassero a ridurre il numero di termini distinti.

2.4 Indicizzazione dei documenti

Arrivati a questo punto, tutte le words (i.e., parole) estratte dai documenti sono state uniformate ma ancora non possono essere esaminate dagli algoritmi perchè devono prima essere trasformate in un formato strutturato, come vettori e matrici. Si applica per questo una procedura di indicizzazione che mappa un documento in una rappresentazione compatta del suo contenuto.

> Tipicamente un documento d_j viene rappresentato come un vettore di pesi $d_j = < W_{1j},..., W_{|A|j} >$ dove $|A|$ è la cardinalità dell'insieme degli attributi con cui si vuole descrivere d_j.

La scelta di quali attributi adoperare deve essere fatta prendendo solo quelli indispensabili e quindi più rilevanti. Esistono diversi metodi per rappresentare i contenuti testuali.

Il modello utilizzato nel progetto di tesi consiste in una rappresentazione matriciale dove ogni colonna rappresenta una particolare feature, e ogni riga corrisponde a un vettore contenente le informazioni necessarie per allenare il modello di Machine Learning, sotto forma di contenuti testuali. Esiste infine una colonna che rappresenta un'etichetta booleana $y \in \{T, F\}$, che denota l'appartenenza o meno a una data categoria, la quale corrisponde a uno degli otto campi con cui vogliamo attuare la fase di categorizzazione.

2.4.1 Riduzione dimensionale

Nella classificazione del testo l'alta dimensione dello spazio dei termini può risultare problematica: gli algoritmi usati per l'induzione dei classificatori non sono efficienti per alti valori di $|A|$. Per questo prima di indurre un classificatore si effettua la riduzione dimensionale: si riduce la dimensione dello spazio vettoriale dei termini $|A|$ a un insieme dei termini ridotto $|A'|$.

L'apprendimento automatico non riguarda solo grandi set di dati. In effetti, non si alimenta il sistema con tutti i dati estratti e preprocessati. Si cerca di alimentare il

sistema con dati attentamente curati, sperando che possa apprendere e forse estendere ai margini la conoscenza che le persone già possiedono. Occorre quindi scoprire quali caratteristiche sono importanti per la previsione e selezionarle per calcoli più veloci e a basso consumo di memoria.

2.5 Machine Learning

Per decenni abbiamo programmato i computer fornendo loro istruzioni in svariati linguaggi. In base alla nostra abilità di formulare al meglio istruzioni da fornire ai computer, siamo stati in grado di realizzare programmi performanti ed efficienti [1]. Sviluppare classificatori automatici o altri sistemi automatizzati sono state progettualità che per lungo tempo son rimaste fuori dalla nostra portata, almeno fino agli inizi del ventunesimo secolo, con l'avvento del Machine Learning.

Il Machine Learning è una tecnologia che abilita le macchine a svolgere una serie di attività senza che queste siano state esplicitamente programmate. Alla base di tutto ciò c'è l'impiego di dataset immensi che consentono di eseguire in modo non deterministico un lavoro basato sul riconoscimento di pattern per mezzo di un computer.

Il concetto di Machine Learning nell'uso della classificazione del testo si riferisce all'approccio di etichettare automaticamente documenti o testi imparando da un insieme di documenti preclassificati. Questo viene fatto selezionando alcune caratteristiche specifiche che devono essere studiate, trovando una correlazione tra loro e da questo prevedere una classificazione grazie a nuovi dati che vengono presentati al modello.

Esistono tre principali categorie di Machine Learning:

- Apprendimento supervisionato: il sistema riceve degli esempi etichettati in base all'output che si vuole ottenere e, a partire da questi dataset, detti di training, deve estrarre una regola generale che associ ad ogni nuovo input l'etichetta corretta.

- Apprendimento non supervisionato: il sistema, a partire dagli input, deve trovare una struttura nei dati, e non esistono dati già etichettati.

- Apprendimento con rinforzo: il sistema riceve input dall'ambiente e attua delle azioni allo scopo di ricevere delle ricompense, che dovrà ottimizzare a seconda dello stato dell'ambiente circostante.

Nel progetto di tesi è stato utilizzato l'apprendimento supervisionato.

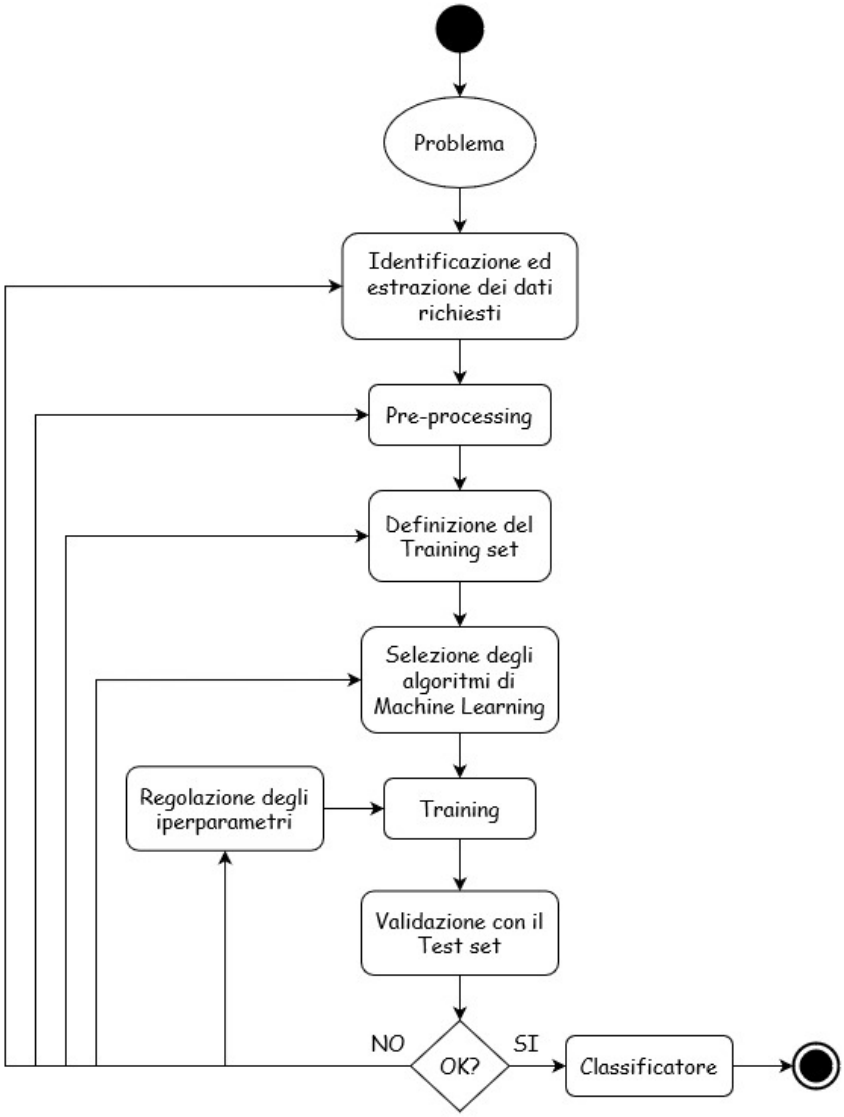

Figura 1: Diagramma delle attività UML dell'apprendimento supervisionato.

Tutti gli algoritmi di Machine Learning rispondono alla medesima logica. L'idea è che si possa rappresentare la realtà mediante una funzione matematica che l'algoritmo non conosce inizialmente ma che può scoprire dopo aver analizzato un certo numero di dati. La realtà e tutta la sua complessità possono essere espresse come funzioni matematiche inizialmente sconosciute, che gli algoritmi di Machine Learning sono in grado però di scovare e utilizzare per uno scopo utile.

Quando si utilizza il Machine Learning attraverso l'apprendimento supervisionato i dati esistenti vengono suddivisi in due parti: una per l'addestramento e una per i test. Il primo set, denominato Training Set, viene utilizzato per "insegnare" al classificatore, osservando le caratteristiche esistenti. Il secondo, chiamato Test Set, viene utilizzato per testare l'accuratezza del modello finale.

L'obiettivo di questo classificatore è quello di assegnare una classe o etichetta a un documento dopo aver esaminato in forma di dati alcune caratteristiche di tale documento. Per assegnare correttamente le classi, il classificatore deve per prima cosa esaminare con attenzione un certo numero di documenti già classificati, ciascuno accompagnato dallo stesso tipo di features. La fase di addestramento richiede che il classificatore osservi molti esempi, in modo che più avanti, nel momento in cui incontra un esempio di documento privo di classe, abbia imparato le regole necessarie a fornire una risposta, riuscendo così ad assegnargli la classe corretta.

Il Machine Learning ha una componente intuitiva e umana molto importante, perchè si tratta di una creazione umana ed è basato sull'analogia in cui gli esseri umani imparano dal mondo che li circonda o con l'imitazione della natura. È importante quindi utilizzare i giusti algoritmi di apprendimento, assegnargli i corretti parametri associati, chiamati iperparametri, e infine fornirgli un adeguato gruppo di documenti dai quali possano apprendere attraverso le features che accompagnano quest'ultimi.

Nel progetto di tesi sono stati analizzati diversi algoritmi di Machine Learning [4], a cui sono stati associati i parametri migliori, e ne verranno presentati tre di questi. Ciascuno di essi ha approcci differenti per risolvere problemi di classificazione e saranno spiegati in dettaglio nella prossima sezione.

2.5.1 Algoritmi

K-nearest neighbors

K-Nearest Neighbors (K-NN) è un algoritmo di apprendimento supervisionato basato sulla vicinanza dei dati [5].

Dato un numero intero positivo k ed un'osservazione del test set x, il classificatore K-NN dapprima identifica i k punti nei dati di training più prossimi a x; poi stima la probabilità che l'osservazione di test appartenga a una determinata categoria j della variabile di risposta come la frazione di punti in k il cui valore di risposta è uguale a j. Infine, classifica l'osservazione x alla categoria con il valore più alto di probabilità stimata.

Necessita quindi di un gruppo di esempi di training già classificati, training set, e quando l'algoritmo analizza i nuovi dati non classificati del test set li classifica in base alla distanza rispetto agli esempi del training set. Per il calcolo della distanza si possono adottare diversi criteri: la distanza euclidea, la distanza Manhattan per i dati numerici e la distanza di Hamming per le stringhe.

Decision Tree

L'albero di decisione, Decision Tree [6], utilizzato per la classificazione del testo è un albero i cui nodi interni sono etichettati in base a dei termini e le foglie sono etichettate in base alle categorie che verranno utilizzate. I rami dell'albero sono determinati dal peso che il termine ha nei dati del test set. Il classificatore classifica il testo passando in modo ricorsivo attraverso le etichette e il loro peso, fino a raggiungere un nodo foglia, raggiungendo così una categoria per la classificazione.

L'algoritmo Decision Tree suddivide i dati in sottoinsiemi composti da attributi diversi, fin quando trova dei sottoinsiemi che raggiungono un medesimo obiettivo con minore incertezza possibile. L'incertezza di un sottoinsieme può essere misurata con diverse metriche, tra i quali l'entropia o l'indice di Gini.

Ensemble Learning

L'apprendimento ensemble prevede l'utilizzo di vari modelli di Machine Learning per migliorare le prestazioni di ognuno di loro preso individualmente [7]. I metodi di questi algoritmi possono essere descritti come tecniche che utilizzano un gruppo di modelli insieme (ensemble), al fine di crearne uno più forte e aggregato.

Esistono diversi tipi di metodi ensemble, a seconda di come vengono combinati tra loro i vari classificatori: Bootstrap aggregating (Bagging), Boosting e Stacking. Nel progetto di tesi sono state utilizzate le cosiddette foreste casuali che fanno parte dei classificatori Bagging.

Il bootstrap è una tecnica di ricampionamento. I dati di training vengono ricampionati $\mathcal{N}$ volte, con $\mathcal{N}$ nell'ordine delle migliaia. Viene costruita una serie di alberi di decisione sugli $\mathcal{N}$ campioni bootstrap.

In questo modo, in una foresta di alberi, possiamo avere alberi con un potere predittivo migliore di quello di un singolo albero. Un albero decisionale troppo grande ha infatti varianza elevata, e grazie al metodo Bagging, che calcola una media sui diversi campioni generati, riusciamo a ridurre la varianza delle osservazioni.

Capitolo 3

Progettazione e implementazione del sistema

3.1 Definizione del problema e analisi dei requisiti

Lo scopo di questo studio è vedere se l'apprendimento automatico può essere utilizzato per semplificare la gestione delle fatture in formato digitale. Diversi algoritmi di Machine Learning sono stati analizzati e utilizzati su testo estratto da motore OCR per confrontare la loro accuratezza e indagare se possono essere considerati fattibili per l'attività.

La tesi mira anche a ricercare la miglior metodologia possibile per estrarre i dati, normalizzarli e indicizzarli al fine di dare in pasto agli algoritmi di Machine Learning dei dataset utili per la successiva classificazione di fatture elettroniche passive.

La motivazione di questa tesi è di aiutare le aziende che gestiscono grandi quantità di fatture, o altri documenti simili, nella gestione e registrazione delle stesse sulla propria piattaforma.

Attualmente, nella maggior parte dei casi, i dati delle fatture vengono registrate manualmente nei sistemi delle aziende e ciò può rappresentare un compito che richiede molto tempo. Pertanto, l'uso dell'apprendimento automatico, potrebbe trasformare i processi di classificazione attuale in un processo molto più efficace in cui i dati vengono classificati automaticamente nei campi necessari per la registrazione, il che porterebbe a grossi tagli nei costi e nel tempo spesi per i task amministrativi.

Ci sono tre diverse domande a cui questo studio si propone di rispondere:

1. L'apprendimento automatico può essere utilizzato per classificare automaticamente le informazioni contenute in una fattura entro o oltre la soglia di precisione ritenuta accettabile dall'azienda in cui è stato sviluppato il progetto di tesi?

2. Quale può essere il miglior modo per indicizzare i documenti delle fatture in base agli otto campi presi in considerazione per la categorizzazione?

3. Quale dei diversi algoritmi di apprendimento automatico analizzati può essere utilizzato per risolvere il compito di classificazione automatica con la massima precisione?

L'ipotesi di questo studio è che l'apprendimento automatico semplificherà e ridurrà gli sforzi manuali richiesti nella gestione e registrazione delle fatture. Ciò significa che i risultati ottenuti da almeno un modello nel caso di studio raggiungeranno un'accuratezza superiore alla soglia che è stata discussa con la Società come miglioramento rispetto al sistema attuale.

A causa della grande quantità di diversi tipi di dati nelle fatture, tutti i campi come importi di denaro, valute e date potrebbero essere difficili da classificare correttamente per un algoritmo di Machine Learning a causa della loro natura non correlazionale.

Per completare lo studio, devono essere raggiunti diversi obiettivi:

1. Analizzare il problema di studio.

2. Raccogliere un numero adeguato di fatture elettroniche passive da cui poter estrarre i dati e ricavare un dataset di documenti preclassificati.

3. Estrarre i dati dai PDF delle fatture raccolte tramite un sistema OCR.

4. Elaborare i dati attraverso una fase di preprocessing.

5. Indicizzare i documenti creando dei dataset opportuni da mandare in pasto agli algoritmi di Machine Learning.

6. Costruire i modelli di Machine Learning utilizzando i dataset creati.

7. Eseguire i diversi algoritmi, addestrarli e testarli sui dataset contenenti i dati presenti sulle fatture.

8. Analizzare i risultati ottenuti dai diversi algoritmi.

Come fase iniziale nel progetto di tesi è stata realizzata un'analisi del problema di studio riguardante due contesti principali: le fatture elettroniche passive e il Machine Learning. Le informazioni raccolte in questa fase hanno lo scopo generale di chiarire, dettagliare e documentare i servizi e le prestazioni che l'applicativo deve offrire e rappresentano il punto di partenza per la progettazione del sistema e per l'intero processo della sua implementazione e validazione.

3.1.1 Analisi delle fatture

Il primo contesto che è stato analizzato è quello concernente la gestione di fatture elettroniche passive e il loro contenuto.

L'azienda Digital Technologies gestisce fatture provenienti da qualsiasi parte del mondo e per la progettazione di un sistema di classificazione è fondamentale comprendere i dati che si ritrovano all'interno, i layout delle fatture utilizzati dai vari fornitori e la loro correlazione in modo da ricercare regole generali per l'estrazione e indicizzazione.

La fattura elettronica altro non è che una fattura in formato digitale, realizzata attraverso un computer, uno smartphone o un tablet che dev'essere fatta seguendo delle linee precise e generali. La fatturazione elettronica passiva in particolar modo è l'attività di ricezione, verifica, archiviazione e conservazione delle fatture ricevute.

Grazie al sistema di gestione e archiviazione di fatture elettroniche passive realizzato dall'azienda, è stato possibile disporre di circa un migliaio di fatture già verificate e classificate grazie alle quali poter avere dei documenti preclassificati e attuare il processo di estrazione e classificazione tramite gli algoritmi di Machine Learning.

La prima fase di analisi delle fatture è stata caratterizzata dal visualizzarne il contenuto e trovare correlazioni tra i diversi termini contenuti all'interno. Solitamente una fattura contiene dati come la data di emissione, un codice identificativo progressivo delle fatture emesse, dati dell'emittente e destinatario, la valuta usata, un elenco di voci con prezzi unitari dei prodotti o servizi ceduti, eventuali tasse e gli importi dell'imposta, l'imponibile e l'importo totale.

Informazioni delle fatture come le date di emissione, le valute o gli importi sono molto sensibili rispetto a un determinato fornitore ed è quindi importante trovare regole generali di estrazione e successivamente normalizzare questi dati nel modo più adeguato possibile.

Nel progetto di tesi sono stati presi in considerazione i seguenti otto campi:

- data di emissione
- numero fattura
- imponibile

- imposta (IVA)
- documento di trasporto (DDT)
- valuta
- importo dell'imposta
- totale dell'importo

Data di emissione

La data di emissione di una fattura varia molto in base al fornitore e al paese di emissione. Se consideriamo come componenti base di un formato della data anno, mese e giorno, possiamo suddividere le date per l'ordine delle componenti base come:

- Big-endian (anno, mese, giorno): ad esempio 2022-02-22
- Middle-endian (mese, giorno, anno): ad esempio 04/22/95
- Little-endian (giorno, mese, anno): ad esempio 22.02.21 o anche 22 febbraio 2021

Oltre all'ordine delle componenti base esistono formati specifici per ogni componente base: aa per l'anno a due cifre, m per il mese a una cifra, mmmm per il mese scritto per intero e diversi altri.

È possibile trovare infine nei formati delle date separatori differenti: la barra obliqua, il punto, il trattino o lo spazio.

Numero fattura

Un numero di fattura è un numero di registrazione univoco assegnato a ogni fattura emessa. Questo numero può essere generato automaticamente da un eventuale software di fatturazione oppure creato manualmente al momento dell'emissione della fattura. Non vi sono normative finanziarie particolari da tenere presenti quando si tratta di creare un sistema di numerazione per le fatture.

Di seguito sono riportate alcune pratiche standard:

- Si utilizza come identificatore le iniziali dell'azienda. Ad esempio, le lettere "DT" potrebbero essere utilizzate per "Digital Technologies", seguite dal numero di fattura, ad esempio DT-0001, che indica la prima fattura emessa a Digital Technologies.

- Si assegnano dei numeri a clienti o aziende. Ad esempio, la prima fattura per il cliente designato con il numero 422 sarebbe 422-0001.
- Si utilizza la data di emissione della fattura, ad esempio 1° gennaio 2021. Questo metodo porterebbe alla numerazione 01012021-0001, 01012021-0002 e così via per tutte le fatture emesse quel giorno.

I numeri di fattura sono utili sia per la contabilità che per il tracciamento dei pagamenti. Sono utili anche per i clienti, che possono farvi riferimento nei propri sistemi di contabilità interna per indicare che particolari fondi assegnati sono destinati a uno scopo specifico, come documentato da una fattura specifica.

Importi

Tra gli importi analizzati ci sono l'imponibile della fattura, l'importo dell'imposta e l'importo totale.

Per imponibile si intende il valore sul quale si applica l'aliquota per determinare l'imposta o il contributo dovuti. I tre importi sono correlati grazie all'IVA: ad esempio, se una sedia costa 60 euro di imponibile più IVA al 22%, l'importo IVA sarà pari a 13,20 euro e l'importo totale sarà di 60+13,20 = 73.20 euro.
Anche gli importi possono avere formati differenti.

È possibile trovare importi composti solo dalla parte intera o composti da una parte intera e una parte decimale.
Anche un importo intero, come ad esempio 180, può avere formati diversi; può essere stato scritto come 180 con solo la parte intera oppure come 180.00 con anche la parte decimale.

Sono da considerare anche i differenti formati dovuti dai diversi separatori utilizzati per separare la parte intera da quella decimale. Solitamente il separatore è la virgola o il punto, ma può capitare che venga utilizzato a volte lo spazio o qualche altro carattere, anche a causa di una sbagliata estrazione dei dati da parte dell'OCR.

Può capitare infine che alcuni fornitori utilizzino lo spazio o il punto per separare le migliaia dalle unità.

È quindi importante cercare di estrarre tutti gli importi trovati nella fattura e normalizzarli in un formato comune.

IVA

IVA è l'acronimo di imposta sul valore aggiunto, ovvero l'imposta che colpisce il valore aggiunto che si produce nel sistema economico per effetto degli scambi di beni e servizi. Più di 140 paesi nel mondo, inclusi tutti i paesi europei, riscuotono l'IVA per gli acquisti al consumo.

Esistono vari tipi di aliquote IVA. L'aliquota varia a seconda del prodotto o del servizio oggetto dell'operazione commerciale. In figura 2 è possibile notare i valori standard delle aliquote nei maggiori paesi Europei nel 2022.

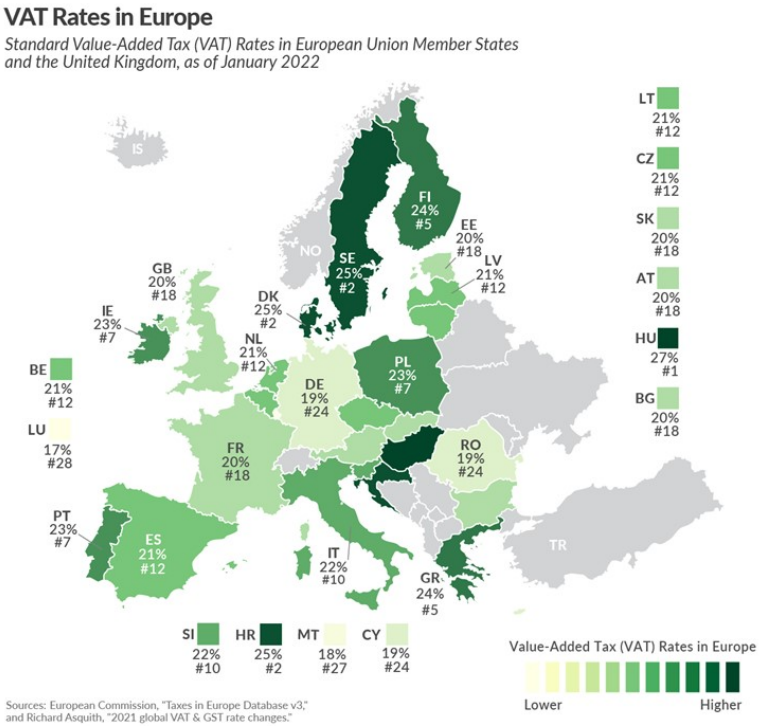

Figura 2: Grafico delle aliquote IVA standard in Europa nel 2022.

DDT

Il documento di trasporto (DDT) è un documento che certifica il trasferimento di merci dal venditore all'acquirente. È stato introdotto dal DPR 472/96 in parziale sostituzione della bolla di accompagnamento.

Il documento di trasporto può essere redatto in forma libera ma deve contenere informazioni importanti quali un numero progressivo, la data di effettuazione e le generalità del cedente e del cessionario.

Il dato che è stato preso in considerazione per la classificazione è la numerazione progressiva, che identifica univocamente il documento di trasporto. I formati dei numeri progressivi del DDT sono in generale gli stessi ma può capitare che in alcuni documenti il numero contenga spazi o altri caratteri. È quindi utile utilizzare delle regole che considerino anche numeri separati da caratteri.

Valuta

La valuta è un'unità di scambio che ha lo scopo di facilitare il trasferimento di beni e servizi.

All'interno di una fattura è possibile identificare la valuta in due modi: tramite codice ISO 4217 o tramite i simboli della valuta. Il codice ISO 4217 è un codice di tre lettere che identifica univocamente una specifica nazione.

L'azienda Digital Technologies gestisce fatture da qualsiasi parte del mondo, dai paesi europei, asiatici o americani. Le valute da valutare sono circa una trentina e per l'estrazione sono stati considerati sia i codici ISO che i simboli delle valute, ciascuno normalizzato successivamente al codice ISO corrispondente.

3.1.2 Analisi delle fatture estratte

La seconda fase dell'analisi delle fatture ha riguardato l'esaminare il modo in cui il sistema OCR estraesse le informazioni dalle fatture.

Prima di estrarre i dati e trasformarli per costruire il dataset, è stato importante poter indagare la logica utilizzata dai sistemi di riconoscimento ottico dei caratteri per estrarre le informazioni dai differenti layout delle fatture. Alcune fatture possono infatti aver avuto problemi di stampa o contenere importanti informazioni scritte in dei caratteri o colori che difficilmente i sistemi OCR possono interpretare.

A questo scopo è stato possibile lavorare con un applicativo Java, realizzato dall'azienda, con cui potessi caricare dei documenti PDF e visualizzarne il documento elaborato, dopo l'upload, dal sistema OCR.
Cliccando su un qualsiasi contenuto testuale del documento, il dato riconosciuto dall'OCR viene visualizzato direttamente in un'apposita casella dell'applicazione. In questo modo, sperimentando su diverse fatture, è possibile direttamente capire se l'estrazione di un determinato dato è avvenuta con successo o ha qualche errore dovuto al rumore dell'immagine, e conseguentemente ragionare su delle regole specifiche di normalizzazione dei dati estratti.

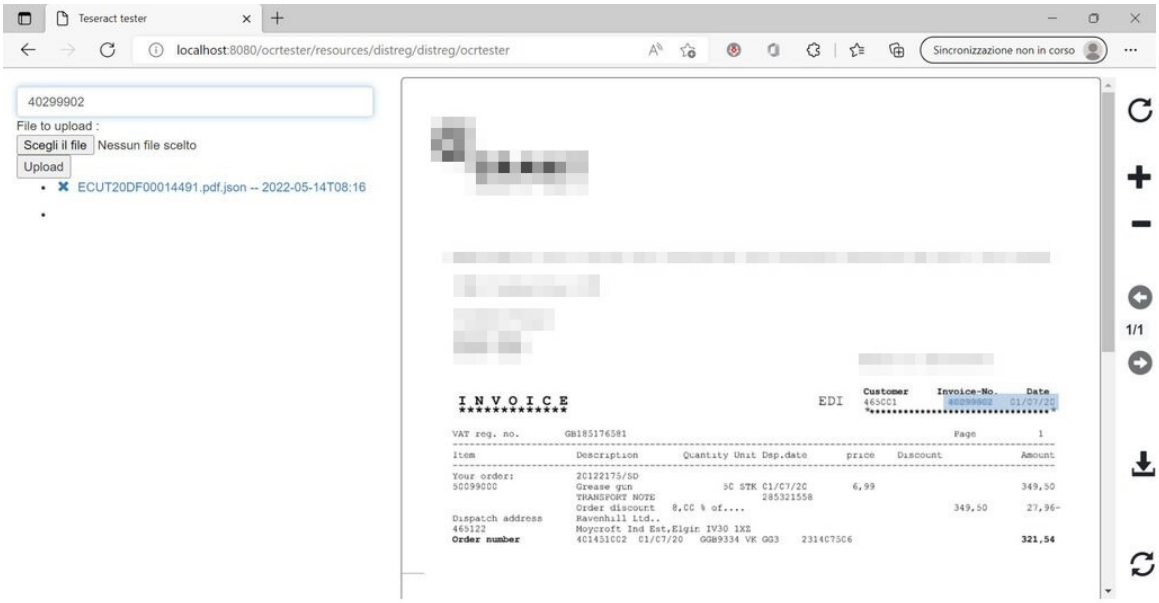

Figura 3: Applicazione Java per l'analisi dei dati delle fatture dopo l'estrazione da parte del sistema OCR.

In figura 3 è possibile osservare l'applicativo Java utilizzato e notare come cliccando sul termine relativo al numero fattura (Invoice-No), appaia il contenuto testuale del dato nella casella in alto a sinistra.

3.1.3 Analisi Machine Learning

Un'altra fase di studio, parallela a quella attuata sulle fatture, è stata l'analisi sul Machine Learning.

La scelta degli algoritmi di apprendimento automatico e dell'approccio da utilizzare per la costruzione dei dataset dipende molto dal dominio di applicazione e dal task di classificazione. Inizialmente la scelta di quale algoritmo utilizzare è ricaduta sulla Regressione Lineare.

Nel Machine Learning la Regressione Lineare è una tecnica di classificazione semplice ma estremamente efficiente per la previsione di valori e classi. L'algoritmo funziona combinando insieme la sommatoria di features numeriche pesate, a cui viene aggiunto un numero costante, chiamato bias.

La formula comunemente usata per illustrare la Regressione Lineare è la seguente:

$$y = \beta X + \alpha \tag{1}$$

In questa espressione, y è il vettore dei valori di risposta. Il simbolo X sta a indicare la matrice delle features da utilizzare per trovare il vettore y. La lettera greca α rappresenta il bias e la lettera β è un vettore di coefficienti utilizzati dal modello di regressione per pesare le features. L'algoritmo deve trovare una relazione tra la matrice delle variabili X e il vettore y.

La limitazione più grave della regressione lineare è che il modello è una sommatoria di termini indipendenti, perchè ciascuna feature rimane isolata nella sommatoria dagli altri elementi ed è moltiplicata solamente per il proprio coefficiente beta. Questa forma matematica è eccellente quando si tratta di esprimere una situazione nella quale le features sono prive di correlazioni tra loro. Quando invece si inseriscono features correlate tra loro nella sommatoria di una regressione, è un pò come se si sommassero le stesse informazioni. A causa di questa limitazione, non è possibile all'algoritmo stabilire in che modo si possa rappresentare l'effetto della combinazione di features sul risultato. Non è quindi possibile rappresentare situazioni complesse in un modello lineare, come quella riguardante la relazione tra le features e le etichette di risposta per i dati delle fatture.

Dopo aver testato l'algoritmo e aver fatto diverse analisi si è optato per scartarlo e scegliere i tre algoritmi descritti nella sezione 2.5.1. La ragione della scelta di questi tre algoritmi si basa sulla loro frequenza di utilizzo in ricerche precedenti e sulle percentuali di classificazione raggiunte. Sono stati effettuati confronti per questi algoritmi specifici durante diversi test e su diversi set di dati. In precedenti ricerche la classificazione del testo è stata effettuata con questi algoritmi su diversi tipi di dati, ma mai in modo specifico su dati estratti dalle fatture.

3.2 Raccolta dei documenti

Gli algoritmi di Machine Learning per poter lavorare hanno bisogno di un grande quantitativo di dati.
A questo proposito, sono stati raccolti circa un migliaio di PDF di fatture elettroniche passive. Sono state scelte delle fatture già analizzate e gestite dall'azienda, in quanto occorreva avere dei documenti preclassificati per poter costruire il modello di Machine Learning.

Un altro aspetto da considerare nella scelta delle fatture è stato quello di scegliere fatture che si differenziassero a causa dei diversi fornitori e paesi di emissione. Un obiettivo importante del progetto è infatti la classificazione dei dati delle fatture a livello globale. Le percentuali di classificazione saranno chiaramente più basse se considerate tutte le differenti fatture, perché cambiano i formati dei dati e i layout delle fatture, ma riuscire ad avere un sistema di categorizzazione automatica di fatture a livello globale rappresenterebbe un grande passo in avanti rispetto ai sistemi attuali delle aziende che gestiscono tali fatture.

3.3 Estrazione dei dati

Per l'estrazione dei dati è stato utilizzato Tesseract. È stato sviluppato un sistema in Java che utilizza la libreria Tess4J [10], un wrapper Java per le API Tesseract che fornisce supporto OCR per vari formati di immagine.

Considerato il fatto che Tesseract supporta i file immagine, i documenti PDF delle fatture raccolte sono state inizialmente trasformate in immagini. Per poter poi gestire queste immagini, il sistema utilizza la libreria Leptonica [11], una libreria open source utile per l'elaborazione e analisi delle immagini. In particolare è stata usata per rimuovere il rumore dalle immagini e le linee all'interno del documento in modo che l'estrazione dei dati venisse compiuta solo su dato testuale e non venisse distratta da altri contenuti che non dovevano essere presi in considerazione. È stata inoltre utilizzata per ruotare determinate fatture qualora fossero state stampate con una qualche angolazione rispetto al formato standard.

Tramite Tesseract si estrae ogni parola contenuta nella fattura che viene trasformata in una struttura dati alla quale si settano informazioni importanti, come ad esempio, oltre al valore del dato, anche le coordinate cartesiane all'interno della fattura e il numero di pagina in cui è stata trovato. Vengono infine raggruppate tutte le parole in un ulteriore struttura dati che sarà usata nella fase di preprocessing.

Per evitare che l'estrazione venisse fatta manualmente per ogni fattura, dato che le fatture da estrarre sono molte, è stata realizzata una procedura tale che prendesse i PDF da una cartella del sistema e facesse l'estrazione per tutte le fatture. Per ogni fattura, terminata la fase di estrazione, viene poi realizzata una fase di preprocessing e infine di indicizzazione.

3.4 Preprocessing

Una volta conclusa la fase di estrazione di tutti i dati presenti all'interno della fattura, occorre selezionare solo i dati di interesse in base ai campi con cui vogliamo attuare la fase di classificazione.

Per gli otto campi presi in considerazione sono stati sviluppati dei metodi in Java che attraverso diverse logiche selezionassero i dati adeguati. La scelta di tali regole di selezione rispecchia i diversi formati dei campi, come descritto nella sezione 3.1.1.

Inizialmente per tutti i campi viene fatta una pulizia dei dati estratti, eliminando dalle parole caratteri che non servono e convertendole in minuscolo per poterle gestire meglio. Successivamente per alcuni campi vengono utilizzate delle stop-words che se corrispondono al dato estratto fanno interrompere l'esecuzione corrente del sistema di selezione dei dati facendo saltare il metodo all'iterazione successiva in modo da lavorare direttamente con un altro dato. Per alcuni campi viene poi realizzata una fase di stemming, in modo da normalizzare i termini estratti. Vengono infine applicate

delle espressioni regolari, diverse per ogni campo, che facciano in modo da estrarre le parole di interesse.

Nella prossima sezione vengono spiegate brevemente le regole di selezione utilizzate per ogni campo.

3.4.1 Data di emissione

Per l'estrazione delle date di emissione, considerati i diversi formati che si potrebbero riscontrare, sono state utilizzate delle logiche per riconoscere l'ordine delle componenti base di una data. Successivamente le date vengono normalizzate in un formato specifico, ovvero il formato dd-MM-yyyy. Infine tutti i separatori vengono trasformati nel carattere "/".

In questo modo si estraggono tutte le date presenti in una fattura che vengono uniformate, pronte per essere utilizzate nella fase di indicizzazione.

Parallelamente alla formattazione, vengono estrapolati i tipi di separatori e pattern della data estratta, prima di normalizzarla. Questi dati saranno importanti per costruire il dataset, in quanto rappresentano delle features che possono aiutare gli algoritmi di Machine Learning a classificare le date.

3.4.2 Numero fattura

Il numero di fattura può essere rappresentato come un semplice numero, come numeri separati da un carattere o come un termine formato da lettere iniziali seguite da un numero.
In una fattura possono esistere molti dati che possono sembrare un invoice number, e per evitare di estrarli tutti sono state usate diverse stop-words.

Come stop-words, per i termini che simboleggiano dei numeri, sono stati usati dei pattern che rappresentano le date e gli importi. Per i termini che invece sono formati da lettere seguite da numeri sono stati usati pattern per evitare di estrarre dati che potessero somigliare all'IBAN e alla partita IVA. Infine se venivano trovate parole precedenti al dato estratto come "shipment", "order" o altre che non c'entrassero col numero fattura, il termine estratto veniva scartato facendo proseguire il metodo all'iterazione successiva.

3.4.3 Importi e IVA

In una fattura possono esistere diversi numeri che non servono nella classificazione, come i dati di residenza dell'emittente o destinatario, ad esempio il numero civico o C.A.P; o ancora, l'elenco di voci con i prezzi unitari dei prodotti.

Tutti questi dati creano problemi quando occorre estrarre gli importi di interesse e l'IVA.

Per l'estrazione di questi campi sono state usate regex per filtrare le parole e identificare i numeri che rispettassero un certo pattern.

Per normalizzare i numeri sono stati usati metodi che trasformassero il dato in un formato comune, ovvero le cifre decimali separate da un punto e nessuna separazione delle cifre che esprimono le migliaia. Qualora un numero non avesse cifre decimali, viene inserito un punto e due zeri dopo le cifre intere.

Per facilitare l'estrazione dell'aliquota, dato che come visto in figura 2 sono valori che si ripetono, è stata usata una lista di stringhe che racchiudessero tutte le partite IVA possibili, in modo che qualora fosse stato trovato un valore uguale a uno della lista, quel termine sarebbe stato estratto.

Per evitare di prendere i valori dei prezzi unitari dei prodotti, sono state usate delle stop-words, come KG o PIECE, che sono state trovate spesso scritte nell'elenco di voci dei prezzi unitari. Sono stati infine usati anche dei pattern delle date come stop-words, in quanto alcune date venivano estratte senza separatori e potevano sembrare degli importi.

3.4.4 DDT

Sono stati usati diversi pattern per l'estrazione della numerazione progressiva del documento di trasporto. È infatti capitato spesso che venissero estratti due o tre numeri separati da spazi che se compattati senza spazio rappresentavano il dato di interesse. Alcuni fornitori utilizzano inoltre una numerazione progressiva utilizzando delle lettere inserite all'interno del numero ed occorre quindi utilizzare regole di estrazione che identifichino anche termini composti da lettere e numeri.

Sono stati usati infine i pattern delle partite IVA e dell'IBAN come stop-words.

3.4.5 Valuta

Per estrarre tutte le valute di una fattura, è stato usato un HashMap che mappasse i codici ISO 4217 delle valute, usate come chiavi, ai simboli corrispondenti, usati come valori. Se un dato fosse corrisposto a uno dei codici ISO o a uno dei simboli sarebbe stato estratto.

Un problema nell'estrazione delle valute è rappresentato dall'errore del sistema OCR nel convertire alcune lettere in simboli delle valute. È capitato che alcune parole contenessero lettere simili ai simboli delle valute e ciò ha reso difficile l'estrazione delle stesse. A questo scopo si è cercato di evitare di analizzare parole di lunghezza superiore a tre, cioè la massima lunghezza dei codici delle valute.

3.5 Indicizzazione dei documenti

Gli algoritmi di Machine Learning lavorano con dati strutturati ed è stato quindi fondamentale trasformare i dati analizzati ed estratti per ogni campo, in una rappresentazione compatta. La fase di indicizzazione permette di creare i dataset necessari per la classificazione dei documenti.

Terminata la fase di selezione dei dati, il passaggio successivo è stato ricavare informazioni importanti per ogni termine estratto. In generale, per ogni campo e per ogni dato, le informazioni raccolte sono state le seguenti:

- un codice identificativo della fattura
- il dato estratto
- le posizioni cartesiane x,y del dato all'interno della fattura
- un indice numerico relativo al dato, che identifica l'ordine in cui è stato estratto nella singola fattura
- l'ultima parola incontrata prima del dato

Per le date di emissione sono stati raccolti anche degli indici numerici che rappresentassero i separatori e i pattern usati.

Una volta ricavate tutte le informazioni per ogni dato di una fattura, quest'ultime vengono assegnate a una struttura dati apposita. Viene poi creato un file CSV per poter costruire il dataset iniziale. Questo file è una tabella le cui colonne rappresentano le cinque features descritte prima e ogni riga viene costruita aggiungendo le informazioni raccolte del dato estratto, ricavate dalla struttura dati. Questo processo viene fatto per ogni campo, ricavando così otto file CSV.

Chiaramente questi dataset non sono ancora pronti per poter essere dati in pasto agli algoritmi di Machine Learning; manca infatti l'informazione più importante, ovvero l'etichetta che denota se il dato estratto è quello corretto o meno.

Per aggiungere l'etichetta è stato utile servirsi delle stesse fatture già classificate. Per ogni fattura sono stati creati dei file CSV con i dati corretti per ogni campo della fattura. Grazie a questi file è stato possibile fare i confronti con gli altri file creati dopo la fase di estrazione e inserire un'altra colonna che indicasse se un determinato dato fosse stato estratto correttamente. È stata inoltre inserita anche la colonna relativa al fornitore della fattura, utile per la fase di classificazione.

Considerato il fatto che potessero esserci più dati corretti in una singola fattura ma che il campo corretto dovesse essere solo uno, sono state usate logiche diverse per ogni campo nell'assegnamento di un solo dato corretto tra quelli contrassegnati come tali. Ad esempio per le date di emissione, è stato assegnato come corretto il primo

dato trovato giusto, in quanto solitamente la data si trova nell'header della fattura; invece per gli importi la logica è stata quella di scegliere l'ultimo dato trovato corretto, perchè si ritrovano spesso nella parte finale del documento.

Questa logica è basata sul fatto che passando agli algoritmi di Machine Learning anche le posizioni cartesiane, possano comprendere meglio quale sia effettivamente il dato estratto correttamente.

3.6 Training e Testing

Una volta creato il dataset completo per ogni campo con i dati necessari, occorre splittarlo in due dataset: un set per l'addestramento e uno per il testing.

E' stato scelto di dividere il dataset in base al fornitore; per ogni record di un determinato fornitore, viene suddiviso il dataset per il 70% nel dataset di training e per il 30% nel dataset di testing, così da avere una ripartizione bilanciata. I due dataset saranno poi utilizzati per addestrare i modelli di Machine Learning e verificarne l'accuratezza.

3.7 Costruzione dei modelli

Per la fase di costruzione dei modelli è stato utilizzato Weka, un ambiente software scritto in Java per l'apprendimento automatico. Weka è un acronimo che sta per Waikato Environment for Knowledge Analysis. È stato sviluppato nell'università di Waikato in Nuova Zelanda, è open source e viene distribuito con licenza GNU General Public License.

Weka permette di poter integrare il proprio codice sorgente [12] in un'applicazione Java-based e di applicare dei metodi di apprendimento automatico ad un set di dati potendone analizzare poi i risultati di accuratezza.

Utilizza un formato particolare per rappresentare i dati che utilizza per il Machine Learning, chiamato ARFF, Attribute-Relation File Format.

Un file ARFF e' costituito da due parti:

- un header contenente le informazioni sugli attributi o features col proprio nome e tipo
- una parte data che contiene i dati nel formato specificato nell'header

Esiste però anche un convertitore per i file CSV che li trasforma in formato ARFF direttamente. Inizialmente i dataset di training e testing vengono infatti convertiti in file ARFF e vengono create delle istanze con cui sarà poi possibile costruire il modello e validarlo. Weka lavora infatti con le cosiddette Instances, un oggetto che

rappresenta un insieme ordinato di attributi, il loro tipo e i loro valori. Le istanze in Weka non sono altro che i dataset, convertiti in ARFF file e trasformati in oggetti con cui è possibile lavorare per costruire poi il modello di classificazione.

Per alimentare il processo di indicizzazione sono state filtrate le istanze utilizzando dei metodi di tokenizzazione di Weka; tramite il metodo NGramTokenizer si è scelto di splittare i dati delle istanze in unigrammi.

Per alcuni campi è stata poi attuata una riduzione dimensionale delle istanze, eliminando delle colonne che sono risultate non efficaci nella fase di classificazione e che disturbavano gli algoritmi di Machine Learning.

Occorre infine settare la cosiddetta classe indice, ovvero l'attributo target delle istanze usata per la classificazione. Nel nostro caso si tratta della feature relativa all'etichetta che denota se il dato estratto è quello corretto o meno.

Una volta che si hanno le istanze pronte, è possibile costruire il modello di classificazione grazie al training set. Per costruire il modello, Weka dà a disposizione diversi algoritmi da poter utilizzare. I tre algoritmi scelti in questo progetto si trovano nei seguenti packages:

- K-nearest neighbors: weka.classifiers.lazy.IBk
- Decision Tree: weka.classifiers.trees.RandomTree
- Ensemble Learning: weka.classifiers.meta.RandomCommittee

Come albero di decisione è stato utilizzato il Random Tree, un albero che considera K attributi scelti casualmente in ogni nodo.

Per la costruzione dei modelli di apprendimento ensemble, è stato usato il Random Committee, un algoritmo che costruisce un insieme di classificatori di base randomizzabili. Ciascun classificatore di base viene creato utilizzando un seme di numero casuale diverso ma basato sugli stessi dati. La previsione finale è una media lineare delle previsioni generate dai singoli classificatori di base. Come classificatore di base utilizza proprio il Random Tree.

Per settare gli algoritmi di Machine Learning coi migliori parametri possibili è stata usata la classe CVParameterSelection, che settata al classificatore permette di ottimizzare gli iperparametri degli algoritmi che gli vengono passati.

3.8 Esecuzione degli algoritmi

Dopo aver costruito il modello di classificazione adeguato, è possibile validarlo con il dataset di testing ed eseguire l'algoritmo di apprendimento automatico sulle istanze da classificare.

A questo punto si passa all'addestramento vero e proprio del modello e successivamente alla valutazione delle prestazioni ottenute calcolando attraverso funzioni predefinite di Weka l'accuratezza raggiunta e la matrice di confusione relativa al test set.

Per l'analisi dei risultati sono state considerate diverse metriche che possano valutare quanto un algoritmo abbia lavorato bene su un determinato dataset. Quando si fa una previsione binaria, ci possono essere 4 tipi di risultati:

- L'algoritmo prevede 0 e la classe è effettivamente 0 : questo è chiamato True Negative , ovvero prevediamo correttamente che la classe sia negativa (0).
- L'algoritmo prevede 0 mentre la classe è in realtà 1 : questo è chiamato False Negative , ovvero prevediamo erroneamente che la classe sia negativa (0).
- L'algoritmo prevede 1 mentre la classe è in realtà 0 : questo è chiamato False Positive , ovvero prevediamo erroneamente che la classe sia positiva (1).
- L'algoritmo prevede 1 e la classe è effettivamente 1 : questo è chiamato True Positive, ovvero prevediamo correttamente che la classe sia positiva (1).

Una metrica per valutare il modello di classificazione può essere la matrice di confusione, che esamina tutte le previsioni fatte dal modello e conta quante volte si verificano ciascuno di questi 4 tipi di risultati. Dal momento che per confrontare due diversi modelli è spesso conveniente avere delle singole metriche, è stata usata come metrica fondamentale per la classificazione la cosiddetta recall, in italiano recupero, metrica che corrisponde al numero di veri positivi diviso il numero di tutti i test che sarebbero dovuti risultare positivi, ovvero veri positivi più falsi negativi. Intuitivamente, questa metrica corrisponde alla proporzione di termini che sono correttamente considerati positivi, rispetto a tutti i termini dati positivi.

Un'altra metrica di valutazione considerata è stata l'AUC, Area under the curve ROC. È utilizzata per combinare la TPR (True Positive Rate), ovvero la recall, con la FPR (False Positive Rate), metrica che corrisponde alla proporzione di termini erroneamente considerati positivi rispetto a tutti i termini dati negativi. L'AUC corrisponde all'area sotto la curva formata dai valori FPR sull'asse delle ascisse e i valori TPR sull'asse delle ordinate. È un valore compreso tra zero e uno e più si avvicina a uno e più il classificatore è ideale.

È stata anche considerata la cosiddetta F-Measure: la metrica tiene in considerazione la precisione e il recupero del test, dove la precisione corrisponde al numero di veri positivi diviso il numero di tutti i risultati positivi, ovvero veri positivi più falsi positivi. Viene calcolata tramite la media armonica di precisione e recupero:

$$F-Measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{2}$$

È sempre un valore compreso tra zero e uno e più è alto, migliore è la classificazione.

Un'ultima metricha scelta per la classificazione è la Kappa statistics (Kappa di Cohen). È un indice di concordanza che tiene conto della probabilità di concordanza casuale ed è usata per misurare la relazione tra le istanze classificate e i veri positivi. È compresa tra -1 e 1 e più si avvicina a uno, più il classificatore ha lavorato meglio della mera casualità.

Capitolo 4

Analisi dei risultati

4.1 Validità dei risultati

L'importanza di uno studio e il risultato che presenta devono avere un certo grado di validità per poter essere accettato come contributo all'ambito di ricerca in cui risiede, o per essere accettato dall'ente o azienda per cui lo studio è stato condotto. La validità di una ricerca ci permette di valutare se quello che è stato trovato rispecchia effettivamente il fenomeno studiato, oppure dipende da variabili di disturbo; ci dice se le deduzioni tratte dai dati sono estensibili ad altri contesti.

Esistono quattro diversi tipi di minacce alla validità, come identificato da Wohlin [9]: interna, esterna, di costrutto e di conclusione.

- Una minaccia alla validità interna di cui tener conto in questa tesi riguarda il cambiamento del contesto che può avvenire nel tempo durante lo studio, causando quindi delle differenze rispetto ai piani iniziali. Un'altra minaccia può essere rappresentata dalla diversità nell'implementazione del progetto o dall'impiego di strumenti poco affidabili.

- Per verificare invece la validità esterna occorrerebbe estendere i risultati dal campione alla popolazione o estenderli a condizioni che non sono sovrapponibili a quelle in cui la ricerca è stata condotta. Bisogna anche considerare la stabilità temporale dei risultati, ovvero il fatto che i risultati tratti da una ricerca rimangano immutati anche in momenti diversi da quello della ricerca.

- Una minaccia alla validità di costrutto è costituita dall'ambiguità delle variabili indipendenti, che si riferisce al fatto che la variabile indipendente da selezionare può non essere quella ipotizzata dal ricercatore. Ogni soggetto della ricerca può avere delle proprie idee soggettive sulla scelta delle features e questo può interferire con la validità dello studio.

- La validità di conclusione descrive la capacità di trarre conclusioni statisticamente corrette in base alle misure. Una minaccia a tale validità può derivare dal fatto di avere un campione troppo piccolo, per cui il test statistico applicato non rileva una relazione significativa. Può essere anche rappresentata dalla presenza di variabili di confusione che aumentano la variabilità di errore, la quale a sua volta influisce sul test statistico usato.

Lo studio nel progetto di tesi ha riguardato un'area di ricerca caratteristica, relativa all'utilizzo di algoritmi di Machine Learning per la classificazione di fatture elettroniche passive. Le regole di selezione dei dati, di preprocessing e di indicizzazione possono cambiare in base al contesto dello studio.

La ricerca si basa inoltre su campioni specifici, il cui contenuto può cambiare sotto diversi aspetti:

- I dati delle fatture possono variare col tempo, non hanno una stabilità temporale.
- Se venissero aggiunte ai dataset fatture di altri fornitori potrebbero essere necessarie altre logiche di estrazione e selezione.
- Considerato il punto precedente, estraendo più dati si avrebbero campioni di maggiori dimensioni e ciò potrebbe inficiare sui risultati degli algoritmi di classificazione.

Un altro aspetto da considerare deriva dall'utilizzo delle librerie per la creazione dei dataset e il successivo processo di classificazione. Può rappresentare una minaccia alla validità in quanto potrebbero essere usati altri strumenti o implementazioni più efficienti, che lavorano meglio su determinati dataset. Occorre considerare anche il fatto che potrebbero esistere algoritmi non implementati in Weka che risultano migliori a livello di performance o sono più funzionali.

L'ultima minaccia alla validità da ponderare concerne la fase di indicizzazione dei documenti. La selezione delle features e le logiche di costruzione dei dataset non sono aspetti propriamente imparziali, ma riguardano osservazioni che tal volta risultano soggettive. Un modo per ridurre tali minacce consiste nella verifica delle correlazioni tra i dati delle variabili che stiamo studiando e le variabili concettualmente simili, o anche verificare se la manipolazione sperimentale è effettivamente rappresentativa del costrutto teorico ipotizzato.

La disponibilità della moltitudine di dati a disposizione può ingannarci e farci credere di aver individuato un modello di validità generale quando, in realtà, abbiamo semplicemente trovato una specifica connessione tra le innumerevoli disponibili. Quando si hanno a disposizione tantissimi dati e potenti macchine per analizzarli

è possibile trovare pseudoevidenze che offrono un'apparente spiegazione per quasi qualunque fatto.

Facciamo analizzare tutti i dati a un algoritmo di supervised learning il quale, imparando da milioni di termini e avendo a disposizione due "vincitori", cerca quella che secondo lui è la regola che contraddistingue il vincitore-tipo. In generale, l'utilizzo dell'apprendimento automatico è legato a filo doppio con la necessità di mettere continuamente in discussione quello che presumiamo di aver capito e chiederci se si può fare di meglio, procedendo anche, se necessario, per tentativi. Dobbiamo allenare la nostra mente alla sperimentazione continua e mantenerla aperta a farsi sorprendere dalle intuizioni dell'intelligenza artificiale.

4.2 Risultati di classificazione

Questo capitolo presenta i risultati acquisiti dai diversi algoritmi per gli otto campi utilizzati nel caso di studio. Saranno presentate le percentuali di classificazione e le metriche di accuratezza degli algoritmi e verranno confrontate le migliori percentuali ottenute per ogni campo. I risultati saranno presentati utilizzando dei grafici a barre per visualizzare la rispettiva accuratezza in percentuale, in termini di True Positive Rate relativo al numero di veri positivi diviso il numero di tutti i test che sarebbero dovuti risultare positivi, ovvero in termini della metrica di recall.

Questa scelta è stata fatta sulla base della volontà del business relativa al contesto di applicazione dello studio di tesi. Lo scopo del progetto è di avere un dispositivo automatico che classifichi in modo corretto i termini contenuti in una fattura, e le performance degli algoritmi utilizzati devono riguardare l'accuratezza nel categorizzare correttamente i dati e non la precisione nell'escludere gli stessi qualora non rappresentassero dei dati da classificare in quel determinato campo. Se considerassimo entrambi i casi, la percentuale di classificazione sarebbe maggiore in quanto i dati da escludere sono di più dei termini corretti e in generale gli algoritmi di Machine Learning utilizzati lavorano bene per ambedue gli scenari.

A riprova di ciò, verrà presentata una tabella comprendente, oltre alle metriche di accuratezza descritte nella sezione 3.8 e i tempi di training e testing, anche la percentuale di Accuracy corrispondente alle osservazioni classificate correttamente rispetto alla classe positiva e negativa, ovvero il rapporto tra la somma dei veri positivi e negativi, e il numero totale delle previsioni effettuate.

4.2.1 K-nearest neighbors

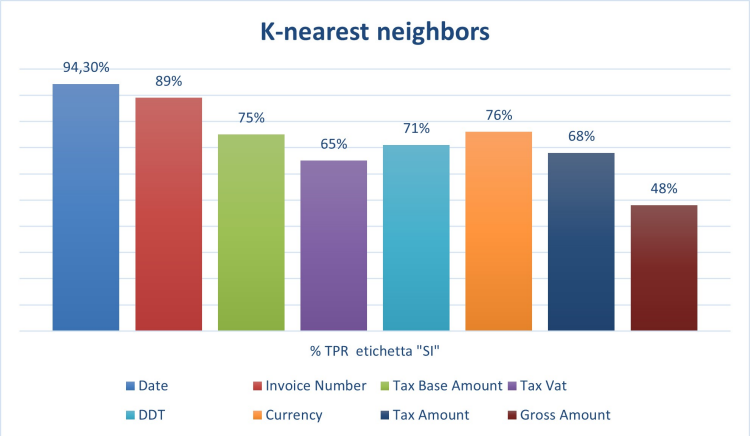

Figura 4: Percentuale di classificazione relativa a recall che si basa sul risultato di previsione corretta, denominato "SI", dei dati delle fatture in base agli otto campi di interesse attraverso l'algoritmo K-NN.

Come si può vedere dalla figura 4, l'algoritmo K-NN ha raggiunto percentuali buone, sopra al 70%, in cinque campi sugli otto presi in considerazione. Fa fatica per i campi numerici come gli importi ma anche per la classificazione dell'IVA.

4.2.2 Random Tree

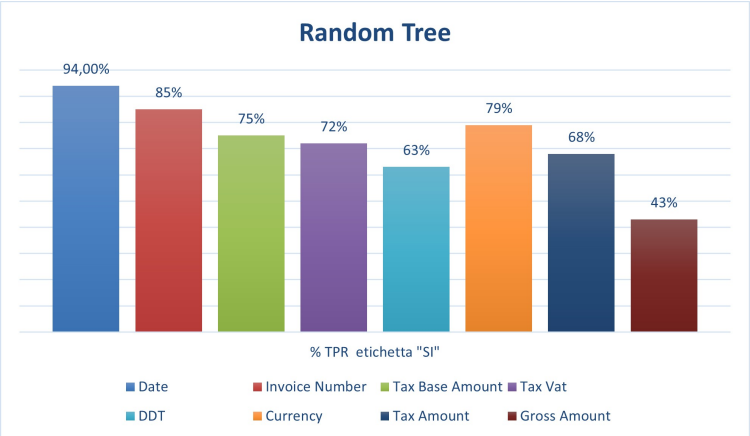

Figura 5: Percentuale di classificazione relativa a recall che si basa sul risultato di previsione corretta, denominato "SI", dei dati delle fatture in base agli otto campi di interesse attraverso l'algoritmo Random Tree.

L'algoritmo Random Tree si è comportato similmente a K-NN, con una maggiore precisione per quanto riguarda la classificazione dell'aliquota ma con un'accuratezza di categorizzazione della numerazione progressiva del documento di trasporto minore e non superiore al 70%.

4.2.3 Random Committee

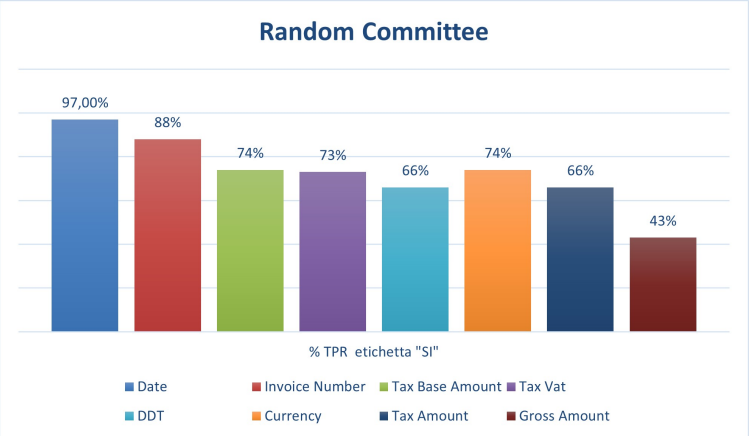

Figura 6: Percentuale di classificazione relativa a recall che si basa sul risultato di previsione corretta, denominato "SI", dei dati delle fatture in base agli otto campi di interesse attraverso l'algoritmo Random Committee.

L'algoritmo Random Committee ha ottenuto risultati migliori per la classificazione delle date di emissione e dell'IVA tra tutti i diversi algoritmi testati. Anche per quanto riguarda l'imponibile e il numero fattura si è comportato bene seppur non superando le percentuali migliori.

4.2.4 Risultati Migliori

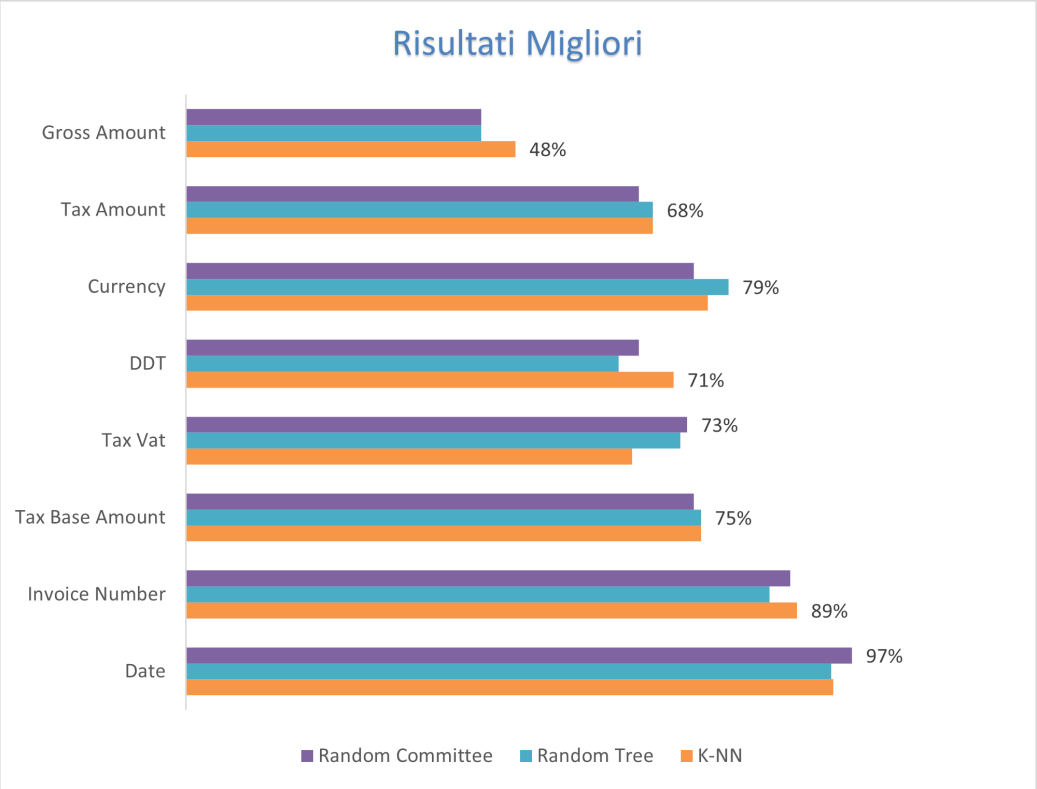

Figura 7: Grafico che riassume le percentuali di classificazione ottenute da ogni algoritmo analizzato con un'etichetta relativa al miglior risultato raggiunto per ogni campo.

Il grafico a barre in figura 7 mostra il confronto dei risultati ottenuti dai tre algoritmi di apprendimento automatico per ogni campo classificato.
È interessante notare come un algoritmo semplice quale il K-nearest neighbors, si comporti efficientemente per diversi campi analizzati.

Complessivamente questi tre algoritmi hanno raggiunto percentuali migliori dei diversi algoritmi di Weka che sono stati testati. Ognuno ha ottenuto percentuali migliori su determinati campi; in particolare gli algoritmi K-NN e Random Tree hanno presentato risultati maggiori per tre campi su otto, mentre Random Committee solo per due campi. Occorre anche aggiungere che i risultati del Random Committee sono abbastanza simili a quelle degli altri due algoritmi, con delle percentuali che rispecchiano la media per tutti gli otto campi.

4.3 Considerazioni

In generale, i tre algoritmi scelti, addestrati e testati hanno ottenuto risultati di accuratezza buoni o molto buoni nella classificazione. I risultati presentati mostrano che su sei degli otto campi classificati, la percentuale di accuratezza supera la soglia concordata dall'azienda all'inizio del progetto, ovvero del 70%.

I due campi di cui la classificazione non può essere considerata accettabile sono quelli relativi agli importi. Una ragione di ciò è probabilmente dovuta all'eccessivo numero di termini numerici all'interno di una fattura. I prezzi degli importi possono essere confusi dai prezzi unitari dei prodotti e la logica di assegnamento dell'etichetta corretta all'ultimo dato correttamente trovato potrebbe essere stata distorta dai diversi layout delle fatture utilizzati dai fornitori; può infatti capitare che una fattura sia composta da una sola pagina e quindi gli importi relativi siano posizionati al termine del documento, validando la logica di assegnamento utilizzata; ma esistono casi in cui le fatture siano strutturate da due o più pagine e gli importi finali vengano registrati all'inizio delle pagine seguenti la prima, lasciando uno spazio bianco al cui termine si possono trovare altri dati numerici che possono confondere gli algoritmi. Un'altra minaccia a questa logica deriva dalle posizioni cartesiane assunte dagli importi estratti, perchè in alcune fatture si possono trovare sulla destra del documento, in altre sulla sinistra o anche in posizione centrale. Risulta infine complicato fare una distinzione tra i vari importi, senza una sorta di approccio basato su regole specifiche per ogni singolo fornitore.

Per quanto riguarda gli altri sei campi, dai risultati ottenuti e analizzati è possibile constatare il fatto che gli algoritmi usati nello studio di tesi potrebbero essere usati per classificare efficacemente i dati delle fatture nelle aziende per il proprio operato. I risultati potrebbero essere ancora più elevati con metodi basati su regole specifiche per ogni fornitore o tramite l'estrazione di informazioni importanti nella fattura che possano aiutare gli algoritmi a predire i risultati. Campi come le valute possono essere classificati come valori di default in base al fornitore, mentre campi più complessi come il numero di fattura o il DDT sarebbero difficili da classificare tramite sistemi predefiniti.

4.3.1 Metriche di accuratezza

	F-Measure	AUC	Kappa Statistic	Training Time (ms)	Testing Time (ms)	Accuracy
Date (Random Committee)	99 %	98.5 %	96 %	464	42	98.8 %
Invoice Number (K-NN)	99.5 %	95 %	90 %	89	17250	99.5 %
Tax Base Amount (Random Tree)	95.7 %	87 %	72 %	169	32	95.7 %
Tax Vat (Random Committee)	97 %	89 %	70 %	1007	74	97.1 %
DDT (K-NN)	99 %	86 %	75 %	71	6455	99.1 %
Currency (Random Tree)	95 %	90 %	80 %	100	25	98 %
Tax Amount (Random Tree)	98.5 %	83 %	75 %	194	42	98.4 %
Gross Amount (K-NN)	97.2 %	75 %	45 %	61	19846	97.1 %

Figura 8: Tabella che mostra le metriche di accuratezza per ogni campo ottenute dagli algoritmi che hanno raggiunto le percentuali migliori.

Nella tabella in figura 8 sono mostrate le percentuali delle metriche di accuratezza descritte nella sezione 3.8. Vengono inoltre visualizzate le tempistiche di costruzione del modello di Machine Learning (training time), e della sua validazione (testing time). È infine rappresentata la percentuale di Accuracy.

Come è possibile notare, le percentuali di F-Measure e AUC sono buone o molte buone per tutti i campi. Ciò indica il fatto che gli algoritmi di classificazione sono riusciti a predire nel miglior modo i veri positivi e negativi. È possibile notarlo anche dalla colonna corrispondente alle percentuali di Accuracy: sono tutte percentuali dal 95% in su, a riprova del fatto che i classificatori hanno lavorato molto bene riuscendo a distinguere quasi tutti i positivi dai negativi.

Per quanto riguarda la Kappa Statistics, per delle percentuali uguali o superiori all'80% la concordanza risulta ottima; per gli altri casi la concordanza è buona o discreta. Il motivo per cui sono state raggiunte percentuali buone o discrete è probabilmente dovuto allo squilibrio tra la TPR ottenuta per le etichette dei positivi e

la TPR per i negativi. Le percentuali mostrate in figura 7 sono infatti peggiori di quelle mostrate nella tabella della figura 8. Nei dataset i dati considerati negativi, e classificati correttamente come tali, sono in numero molto maggiore rispetto ai dati positivi, e per questo le percentuali migliorano.

Infine è utile osservare le differenze dei tempi di training e testing per i diversi algoritmi. I tempi sono stati misurati in millisecondi. Il training time considera il tempo trascorso dall'inizio alla fine della costruzione del modello grazie al training set; il testing time considera invece il tempo di validazione del modello tramite il test set.

Si può notare come nel caso dell'algoritmo K-NN il training time sia molto basso. È infatti chiamato anche algoritmo di apprendimento pigro perché non esegue alcun addestramento quando gli vengono forniti i dati di training. Memorizza solamente i dati durante il tempo di addestramento non eseguendo alcun calcolo e non crea un modello affinché non gli viene passato il dataset di test.

Poichè lo scopo del progetto di tesi è vedere se uno qualsiasi degli algoritmi è adatto per essere implementato in un'applicazione del mondo reale, è importante esaminare il testing time per i diversi algoritmi, considerando che ciò che l'algoritmo dovrebbe fare in un'applicazione sarebbe ricevere i dati da una fattura e infine classificarla.

Guardando i tempi di test, non sembra esserci alcuna differenza evidente tra nessuno degli algoritmi, a parte K-NN, il cui tempo di test supera di gran lunga gli altri. L'algoritmo più veloce è il Random Tree.

Capitolo 5

Conclusioni e sviluppi futuri

5.1 Conclusioni

Questa tesi mirava a ricercare se il Machine Learning fosse adatto a classificare i campi di testo di fatture elettroniche passive con un'accuratezza accettabile o meno. La motivazione è aiutare le aziende con grandi quantità di fatture digitali ad automatizzare i processi in cui viene effettuata la classificazione dei dati. Sono stati concordati livelli di accettazione tali per ogni campo in quanto potrebbero velocizzare i processi di grande rilievo.

Diversi algoritmi di apprendimento automatico sono stati addestrati e testati sui dati contenuti nelle fatture. I modelli sono stati creati in Java con una libreria utilizzata per l'apprendimento automatico chiamata Weka. Ogni algoritmo ha lavorato sugli stessi dataset.

Dopo aver addestrato e testato gli algoritmi, e aver acquisito i risultati, sono stati scelti gli algoritmi che hanno ottenuto le migliori percentuali, che sono stati presentati e descritti nella tesi.

I risultati acquisiti dai tre algoritmi hanno mostrato che la maggior parte dei modelli ha superato la soglia accettabile concordata per sei campi sugli otto analizzati. Ciò dimostra che l'apprendimento automatico può fornire una riduzione dello sforzo necessario per classificare le informazioni sulle fatture.

Un altro aspetto da considerare riguarda la validità dei risultati ottenuti. La validità della tesi dipende dalla gestione delle minacce di validità individuate nella sezione 4.1.

Una minaccia identificata era correlata all'affidabilità nell'utilizzo della libreria Weka per la classificazione dei dati delle fatture.

Weka si è dimostrata rapida e affidabile nei calcoli e nella presentazione dei risultati. La quantità di funzioni e algoritmi presenti, unitamente a quelli presenti nel Package Manager di Weka da scaricare e importare, e a quelli che saranno disponibili

a breve termine tramite i continui aggiornamenti, estendono il potenziale campo di applicazione delle tecniche di data mining a nuove tipologie di dati e raffinano quelle già esistenti. I risultati numerici ottenuti dagli algoritmi sono inoltre precisi e ricchi di dettagli.

Un'altra minaccia individuata era associata al contenuto delle fatture e al suo possibile cambiamento nel tempo.

A questo scopo sono state raccolte fatture che cercassero di racchiudere il meglio possibile i diversi formati dei dati contenuti in una fattura e i differenti layout delle fatture utilizzate dai fornitori. Sono state inoltre utilizzate regole di selezione dei dati tali da identificare e filtrare qualsiasi tipo di formato incontrato. L'aggiunta di formati dei dati differenti nelle fatture comporterebbe solo l'inserimento di ulteriori regole di selezione, senza sconvolgere l'implementazione del sistema.

Per la gestione dell'ultima possibile minaccia riscontrata, pertinente la validità in fase di indicizzazione, sono stati fatti diversi esperimenti con features differenti in modo da trovare la giusta correlazione tra i dati e i campi da classificare. Ciò ha riguardato in primo luogo la riduzione dimensionale dei dataset da classificare, in quanto alcune features riducevano la percentuale di classificazione, come è capitato a volte per la colonna relativa all'ultima parola trovata prima del dato estratto. Ha riguardato inoltre l'estrazione di particolari informazioni, inserite come features all'interno dei dataset, che per alcuni campi hanno aumentato le percentuali di categorizzazione, come ad esempio l'informazione sui separatori e pattern usati nelle date di emissione.

Anche se il livello di accuratezza raggiunto potrebbe non essere completamente generalizzabile a tutte le aziende che gestiscono fatture o altri dati digitali, e avere minacce di validità dovute all'instabilità temporale dei risultati o all'utilizzo di strumenti e campioni specifici, lo studio di questa tesi potrebbe essere visto come un punto di partenza per quando l'automazione nel processo di classificazione automatica di fatture potrà prendere sempre più piede e migliorare la gestione di quest'ultime.

Il vantaggio sociale dei risultati trovati in questa tesi riguarda la possibilità di risparmiare tempo e denaro in molte aree in cui le fatture vengono gestite in formato digitale. Queste aree sono, di giorno in giorno, in aumento di numero. Anche industrie che gestiscono documenti simili alle fatture possono trarre vantaggio da questa ricerca. Date, importi, valute e altri dati utilizzati come categorie in questo studio, compaiono in gran parte del mondo digitale e possono avere un impatto su industrie di ogni tipo quando la gestione deve essere eseguita manualmente. Gli sviluppi futuri basati sui risultati di questa tesi potrebbero automatizzare completamente la categorizzazione dei campi delle fatture, il che a sua volta potrebbe portare a un processo molto più efficiente che influenzi positivamente le società che ne fanno uso.

Anche se la categorizzazione completamente automatizzata è una possibilità in futuro, molto probabilmente sarà sempre necessaria una sorta di gestione vigilata. Se

la classificazione categorizza i dati in modo sbagliato, c'è il rischio che dati personali o finanziari vengano archiviati in modo errato. Le implicazioni di una gestione sbagliata possono essere potenzialmente rischiose e vi è uno svantaggio etico per la potenziale automatizzazione della classificazione dei dati delle fatture. Ci sono attualmente un certo numero di persone che lavorano classificando manualmente i dati e se il loro compito dovesse essere automatizzato, c'è il rischio imminente che il loro tipo di impiego venga considerato superfluo. Anche se un piccolo numero di supervisori dovessero essere considerati necessari per evitare errori di classificazione, molti altri potrebbero non esserlo e ciò avrebbe un impatto negativo sulla società in generale. Se non è possibile arrestare l'avanzata del machine learning, resta imprescindibile la necessità di cambiare le dinamiche nel mondo del lavoro. L'unico modo per arginare il senso di "minaccia" consiste nell'avere più lavoratori formati nella gestione dell'intelligenza artificiale, per non subirla passivamente [2].

5.2 Sviluppi futuri

Per poter portare la ricerca svolta in questa tesi a un passo successivo e farla funzionare al meglio al di fuori di questo studio, occorrerebbe inserire logiche specifiche per la classificazione di determinati dati.

Un grosso problema che è stato riscontrato riguarda la categorizzazione degli importi, in particolare il riconoscimento dello specifico tipo di importo. Progetti futuri potrebbero integrare dei template appositi per riconoscere, in ogni fattura di un particolare fornitore, la posizione che solitamente assume l'importo che interessa estrarre, in modo che per le successive estrazioni il sistema saprà già dove dover guardare all'interno della fattura. Questi template potrebbero utilizzare anche delle keywords che richiamino la collocazione della sezione nella fattura dove vengono posizionati gli importi, in modo che anche se i layout dei fornitori cambiano col tempo, le parole chiave possano diventare delle "ancore" che aiutano il sistema nell'estrazione.

L'inserimento di template in base ai fornitori può provocare però anche l'aggiunta di ulteriori regole di selezione, perchè non è detto che uno stesso fornitore di una stessa società non cambi mai il proprio layout per le fatture. Il cambiamento dei layout comporta a sua volta la modifica o inserimento di determinate logiche per quella particolare azienda.

Per altri campi come le valute sarebbe invece comodo includere nel sistema un meccanismo per assegnare dei valori di default in base ai fornitori o a fatture precedenti. Per avere un tale sistema si potrebbe connettere l'applicativo automatizzato a un database, in modo che i dati di fatture precedenti possano diventare informazione ulteriore per la loro ricerca e successiva estrazione in nuove fatture.

Un altro sviluppo che si potrebbe apportare riguarda il miglioramento nell'estrazione da parte del motore OCR. Una buona estrazione del testo si ha quando l'immagine o il PDF della fattura non hanno avuto difetti di stampa e il motore può lavorare senza confondere certi caratteri con altri. Qualora ciò non accada, modificare al meglio l'immagine prima di darla in pasto alla fase di estrazione potrebbe portare a dei vantaggi nella classificazione automatica.

Un ultimo punto che potrebbe migliorare le percentuali di classificazione e anche le performance del sistema è inerente alla scelta dei diversi parametri da utilizzare per gli algoritmi di apprendimento automatico. Con uno studio più approfondito e con una maggiore esperienza, si potrebbero utilizzare degli iperparametri specifici per il singolo algoritmo e per il dataset da classificare.

Bibliografia

[1] L. Massaron, J.P. Mueller, Machine Learning For Dummies, Milano, Hoepli, 2019.

[2] L. Massaron, J.P. Mueller, Intelligenza Artificiale For Dummies, Milano, Hoepli, 2020.

[3] A. Larsson, T. Segeras, Automated invoice handling with machine learning and OCR, Stoccolma, 2016.

[4] R. Ratra, P. Gulia, N.S. Gill, Performance Analysis of Classification Techniques in Data Mining using WEKA, India, 2021.

[5] V. Bijalwan, V. Kumar, P. Kumari, J. Pascual, KNN based Machine Learning Approach for Text and Document Mining, India, 2014.

[6] Mr. Brijain, R. Patel, Mr. Kushik, K. Rana, A Survey on Decision Tree Algorithm For Classification, India, 2014.

[7] R. Polkar, Ensemble based systems in Decision making, New Jersey, 2006.

[8] F. Sebastiani, Machine learning in automated text categorization, Italia, 2002.

[9] C. Wohlin et al., Experimentation in Software Engineering, Svezia, 2012.

[10] Tess4J - JNA wrapper for Tesseract. `http://tess4j.sourceforge.net`.

[11] Leptonica Library. `https://github.com/danbloomberg/leptonica`.

[12] Weka Source Code. `https://svn.cms.waikato.ac.nz/svn/weka/branches/stable-3-8`.