ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il management

SVILUPPO DI UN SISTEMA DI CROWDSOURCING PER LA VALIDAZIONE E L'ARRICCHIMENTO DI DATASET

Relatore:

Chiar.mo Prof.
Paolo Ciancarini

Presentata da:

Hamza Elatfi

Sessione I
Anno Accademico 2018 / 2019

Indice

Capitolo 1

Introduzione

Il presente lavoro ha come oggetto lo sviluppo di un software che permetta di validare annotazioni di un dataset precaricato ed estendere quest'ultimo con nuove entità e annotazioni in maniera semplice e del tutto gratuita. Questo progetto risponde alla necessità di costruire dataset annotati di buona qualità attraverso la tecnica di crowdsourcing. Questi dataset sono fondamentali per lo sviluppo di algoritmi sempre più efficienti nell'ambito del machine learning.

La motivazione che mi ha spinto ad approfondire tale tema è il forte interesse verso le tecnologie attuali di intelligenza artificiale. Tecnologie che utilizziamo tutti i giorni durante le nostre attività senza sapere i meccanismi e le logiche che ci stanno dietro. Un esempio di strumenti che utilizziamo nella vita di tutti i giorni sono: ricerche suggerite nei browser, compilazione automatica di parole, assistente vocale di Goole, Siri, Alexa, ecc… Tutti questi strumenti dotati di intelligenza artificiale sfruttano il machine learning per imparare e migliore il proprio utilizzo. Il progetto sviluppato in questa tesi, ricade nel ramo del machine learning supervisionato, argomento che verrà trattato nei capitoli successivi.

L'obbiettivo di questa tesi di laurea è creare un sistema che permetta di ottenere dei feedback su dataset già annotati e di far inserire ai crowdworker nuove entità da far valutare ad un algoritmo di classificazione. Questo sistema permette, a livello

accademico, di migliorare le annotazioni in maniera semplice e gratuita così da evitare metodi più costosi. Questi metodi vengono utilizzati, invece, nel settore industriale in quanto vi è maggiore disponibilità sia in termini economici che in termini di forza lavoro. Un esempio può essere Google, che utilizza i propri utenti per migliorare la qualità dei propri dataset annotati proponendo delle immagini da far valutare.

La tesi è articolata in quattro capitoli:

Cap. 2. Intelligenza artificiale e machine learning

Nel secondo capitolo verranno introdotte le nozioni base con relativi esempi, utili per comprendere l'argomento della tesi e del sistema creato.

Cap. 3. Crowdsourcing

Nel terzo capitolo vi sarà una spiegazione dettagliata della tecnica maggiormente utilizzata nel campo del machine learning. Tale tecninca verrà utilizzata anche nel sistema creato per raggiungere gli obbiettivi prefissati.

Cap. 4. Sistema di crowdsourcing sviluppato

Questo capitolo tratterà le analisi effettuate nella fase di progettazione del sistema, ovvero analisi dei casi d'uso e di dominio. Dopodichè ci sarà una spiegazione della composizione dell'architettura finale del sistema e del funzionamento di quest'ultimo nello svolgimento delle principali operazioni.

Cap. 5. Conclusioni

Il capitolo finale conterrà delle considerazioni finali e dei possibili sviluppi del software creato.

Capitolo 2

Intelligenza artificiale e Machine Learning

L'argomento trattato nell'elaborato appartiene al ramo del "Machine learning" che a sua volta fa parte della materia generale che possiamo indicare come intelligenza artificiale. In questo primo capitolo verranno introdotte delle nozioni generali per aiutare a comprendere ed inquadrare meglio l'argomento in questione.

2.1 Che cosa è l'intelligenza artificiale?

L'Intelligenza Artificiale[S1] è quella disciplina dell'informatica che studia i fondamenti teorici e i metodi per costruire macchine intelligenti, ovvero capaci di risolvere problemi e riprodurre attività proprie dell'intelligenza umana.
Sebbene l'uomo abbia da sempre cercato di comprendere e riprodurre la mente umana e i suoi processi cognitivi, gli albori dell'Intelligenza Artificiale come scienza si collocano nel dopoguerra. Nell'anno 1956 essa ottiene lo status di vera e propria disciplina scientifica: in quell'anno si tenne infatti la conferenza del Dartmouth College, organizzata dal ricercatore John McCarthy[1], in cui verrà proposto il nome "Intelligenza Artificiale", e ne verrà esposto il primo programma a opera di Simon e Newell[2]. Lo

[1] John McCarthy, informatico statunitense inventore del termine "intelligenza artificiale".
[2] Simon e Newell, creatori di uno dei primi programmi di intelligenza artificiale chiamato "General Problem Solver".

sviluppo è segnato da alti e bassi, dovuti alle grandi aspettative nate dall'ottimismo dei primi ricercatori e dall'immensa difficoltà nella creazione di una macchina che pensa e ragiona come noi. Dopo un'iniziale delusione, l'IA si è mossa più realisticamente, tendando di risolvere problemi di complessità inferiore con grande successo. Ed è a questo punto che occorre distinguere due forme di intelligenza artificiale: debole e forte, così denominate dallo studioso John Searle[3].

L'Intelligenza Artificiale debole si pone l'obbiettivo di simulare l'intelligenza umana. È utilizzata per svolgere complesse funzioni umane, ma non è in grado né di comprendere, né di pensare come un essere umano: si limita perciò a svolgere egregiamente un compito umano, senza mai eguagliarlo né superarlo. In questo ambito, la tecnologia ha raggiunto grandi traguardi. Nel 1997, il supercomputer Deep Blue, sviluppato da IBM, sconfisse il campione del mondo di scacchi Garry Kasparov; e ancora nel 2016, Google DeepMind AlphaGo sconfigge Lee Sedol, 18 volte campione del mondo di Go[4]. Da allora i computer hanno fatto ulteriori passi da gigante, superando infinitamente le abilità umane in determinati compiti.

L'Intelligenza Artificiale forte, d'altro canto, se programmata opportunamente, diventa essa stessa una mente, con una capacità cognitiva indistinguibile da quella umana. Per fare ciò, si adopera la tecnologia dei sistemi esperti, ovvero una serie di programmi che vogliono replicare le competenze di una persona esperta in un determinato ambito. È questo l'ambito dell'Intelligenza Artificiale con più potenzialità, che potrà arrivare a creare i robot che osserviamo sui grandi schermi. Da decenni si dibatte se la realizzazione di una tale macchina sia possibile, o meglio ancora desiderabile. Se supererà le nostre capacità, chi dice che saremo in grado di controllarla? Questa super intelligenza finirà con il ritenerci una minaccia? O semplicemente d'intralcio?

Sono questi gli interrogativi tecnici e filosofici che circondano l'Intelligenza Artificiale, un ambito che si è già dimostrato rivoluzionario, sia dal punto di vista tecnologico-industriale, che nella sfera filosofico-morale.

---

[3] John Searle, principale critico dell'intelligenza artificiale in senso forte, non considera assimilabile il computer alla mente umana.

[4] Gioco da tavolo strategico per due giocatori, l'obbiettivo è controllare una zona maggiore di quella dell'avversario.

L'Intelligenza Artificiale ha bisogno di 4 componenti per funzionare efficacemente:

- **Natural Language Processing**: è la componente che rende possibile la comprensione del nostro linguaggio alla macchina: attraverso appositi algoritmi, il computer riesce a identificare le varie parti del discorso e comprenderne il significato;
- **Knowledge Representation**: è il campo dell'intelligenza artificiale che si occupa di organizzare le informazioni in modo tale da permettere alla macchina di risolvere compiti complessi (es. dialogare con un essere umano, diagnosticare una malattia);
- **Automated Reasoning**: utilizza i dati presi in input e successivamente organizzati per produrre output e conclusioni;
- **Machine Learning**: usa la statistica e le leggi probabilistiche per trovare nuovi schemi decisionali ed essere in grado di adattarsi alle circostanze.

2.2 Che cosa è il Machine Learning?

Il Machine Learning è uno degli ambiti fondamentali dell'Intelligenza Artificiale. Il Machine Learning non è semplice da spiegare in termini elementari, dato l'elevato numero delle tecniche di apprendimento, modalità e tecniche utilizzate, che danno origine ad altrettante applicazioni e impieghi.

Potremmo definirlo come un metodo di analisi dati che consente la costruzione autonoma di modelli analitici. Questo metodo si basa sull'idea che i sistemi sono in grado di imparare dai dati, identificando dei modelli e prendendo decisioni con il minor intervento umano possibile. In altri termini, il Machine Learning è l'insieme delle tecniche e dei meccanismi che consentono a una macchina intelligente di migliorare le proprie capacità e prestazioni nel tempo.

A seconda del metodo con cui una macchina impara e accumula dati, possiamo distinguere tre differenti sistemi di apprendimento automatico:

1. **Apprendimento supervisionato**: consiste nel fornire al computer, sotto forma di codice, una serie di dati e modelli per costruire un vero e proprio database di informazioni ed esperienze. In tal modo, la macchina dovrà essere in grado solamente di attingere ai dati preconfezionati una volta stimolata, e rispondere così nella maniera più opportuna;
2. **Apprendimento non supervisionato**: questo metodo prevede che le informazioni inserite non siano sotto forma di modelli: per questo, la macchina non conosce le conseguenze di una tale scelta. Spetterà dunque al computer organizzare i dati, e in base alle esperienze svolte, adattarsi passo dopo passo e migliorare sempre di più le proprie risposte agli stimoli;
3. **Apprendimento per rinforzo**: questa metodologia prevede che la macchina sia fornita di sistemi e strumenti che le consentano di migliorare le proprie capacità di apprendimento e di adattarsi all'ambiente circostante. Questo sistema è tipico delle auto senza pilota, che grazie a sensori di supporto riesce a rilevare gli ostacoli circostanti e adattare il proprio percorso costantemente.

Le applicazioni di queste tecniche sono sconfinate, tanto che diverse fanno parte della nostra vita di tutti i giorni senza che ce ne si renda conto.
Un primo, chiaro esempio è quello dei motori di ricerca, che grazie all'inserimento (input) di poche parole chiave restituiscono (output) una lista di collegamenti ritenuti pertinenti alla nostra ricerca a seguito di un'analisi dei dati accessibili alla macchina.
Un altro esempio è quello dei filtri anti-spam. Grazie al Machine Learning costante, la macchina impara a intercettare i messaggi sospetti e fraudolenti.
In campo tecnologico, la realizzazione di auto a guida autonoma è possibile solo grazie al Machine Learning costante dei dispositivi, che svolgono un'attenta analisi dell'ambiente circostante grazie ai sensori a loro disposizione e pianificano il percorso e la velocità conformi alle situazioni.
I social media, così come i siti di eCommerce e di intrattenimento, basano parte del loro business sull'anticipazione dei bisogni e dei desideri degli utenti, possibile grazie a

sistemi di Machine Learning: queste piattaforme analizzano le abitudini dei consumatori e suggeriscono prodotti che potrebbero soddisfare i gusti personali dei clienti.

2.3 Principali algoritmi di Machine Learning

L'intelligenza artificiale è in grado di utilizzare i dati per insegnare a se stessa grazie agli algoritmi di machine learning chiamati anche algoritmi di apprendimento. Quest'ultimi forniscono un modello che l'IA applica ad un set di dati per risolvere il problema. Di seguito verranno descritti i principali algoritmi di apprendimento che vengono utilizzati nell'ambito dell'IA.

2.3.1 Naïve Bayes Classifier Algorithm

Il Naïve Bayes Classifier è un algoritmo di classificazione dei contenuti testuali. Un classificatore è una funzione che assegna il valore di un elemento di una popolazione da una delle categorie disponibili. Questo classificatore è uno dei metodi di apprendimento più popolari raggruppati per similarità, che funziona sul famoso teorema di probabilità di Bayes[5], per costruire modelli di apprendimento automatico, in particolare per la previsione di eventi e la classificazione dei documenti.

Principali vantaggi dell'applicazione dell'algoritmo classificatore di Naïve Bayes:

1. L'algoritmo è molto efficace quando le variabili di input sono categoriali.
2. Un classificatore di questo tipo converge più velocemente, richiedendo dati relativamente piccoli rispetto da altri modelli, quando è in vigore l'ipotesi di indipendenza condizionale di Naïve Bayes[6].

[5] Il teorema di Bayes deriva da due teoremi fondamentali delle probabilità: il teorema della probabilità composta e il teorema della probabilità assoluta. Viene impiegato per calcolare la probabilità di una causa che ha scatenato l'evento verificato.

[6] Per indipendenza condizionale si assume che la presenza o l'assenza di una particolare feature non è correlata alla presenza o assenza di altre feature.

3. Con questo algoritmo è più facile prevedere la classe del set di dati del test oltre ad avere una buona previsione delle multi-classi.
4. Sebbene richieda l'assunzione di indipendenza condizionale, il classificatore ha presentato buone prestazioni in vari domini applicativi.

Principali applicazioni del classificatore di Naïve Bayes:

- *Analisi dei sentimenti:* viene utilizzato su Facebook per analizzare gli aggiornamenti di stato che esprimono emozioni positive o negative.
- *Categorizzazione dei documenti:* Google utilizza la classificazione dei documenti per indicizzare i documenti ed assegnare ad ognuno di essi un punteggio in base alla sua importanza, es. PageRank.
- *Classificazione delle notizie su tecnologia, intrattenimento, sport, politica, ecc.*
- *Filtro di spam:* Google Mail utilizza l'algoritmo per classificare le email come spam o non spam.

2.3.2 K-Means Clustering Algorithm

K-Means è un algoritmo di apprendimento automatico non supervisionato utilizzato comunemente per l'analisi dei cluster[7]. K-Means è un metodo non deterministico e iterativo che opera su un dataset di dati attraverso un numero predefinito di cluster (k). L'output dell'algoritmo è k cluster con dati di input suddivisi tra i cluster.

Principali vantaggi dell'algoritmo K-Means:

1. In caso di cluster globulari, l'algoritmo produce cluster più ristretti rispetto al raggruppamento gerarchico.
2. Dato un valore inferiore di k, l'algoritmo calcola più velocemente, rispetto ai clustering gerarchici, per un numero elevato di variabili.

[7] Gruppo di dati.

Mediante questo algoritmo, utilizzato dalla maggior parte dei motori di ricerca, vengono raggruppate le pagine Web per similarità e "tasso di rilevanza". Questo aiuta i motori di ricerca a raggruppare le pagine web in base a ciò che cerca l'utente e al tempo stesso diminuire i tempi di calcolo.

2.3.3 Support Vector Machine Learning Algorithm

Support Vector Machine è un algoritmo di apprendimento automatico supervisionato per problemi di classificazione o regressione in cui il set di dati insegna all'algoritmo quali sono le classi possibili, in modo tale che possa classificare qualsiasi nuovo dato. L'algoritmo funziona classificando il dataset in classi diverse individuando una linea (hyperplane) che separa il dataset di allenamento in classi. Poiché esistono molti hyperplane, l'algoritmo SVM cerca di massimizzare la distanza tra le varie classi coinvolte e questo viene identificato come massimizzazione del margine. Se viene identificata la linea che massimizza la distanza tra le classi, aumenta la probabilità di generalizzare in maniera ottimale i dati non visibili.

Gli SVM si dividono in due categorie:

- **SVM lineari** – i dati di addestramento sono separati da un hyperplane.
- **SVM non lineari** – non è possibile separare i dati di allenamento usando un hyperplane in quanto i dati sono troppo complessi per cui è impossibile trovare una rappresentazione per ogni vettore di funzionalità (es. riconoscimento dei volti).

Principali vantaggi dell'utilizzo di SVM:

1. SVM offre le migliori prestazioni di classificazione sui set di allenamento.
2. SVM rende efficiente la corretta classificazione dei dati futuri.
3. Non fornisce alcuna forte ipotesi sui dati.
4. Non si adatta eccessivamente ai dati.

Questo algoritmo viene comunemente usato per le previsioni del mercato azionario da varie istituzioni finanziarie. Viene utilizzato per confrontare le performance di titoli azionari dello stesso settore così da aiutare a gestire le decisioni di investimento in base alle classificazioni fatte dall'algoritmo.

2.3.4 Apriori Machine Learning Algorithm

L'algoritmo Apriori è un algoritmo di apprendimento automatico non supervisionato che genera regole di associazione da un determinato set di dati. La regola di associazione implica che se si verifica un elemento A, allora con buona probabilità si verificherà anche l'elemento B. La maggior parte delle regole di associazione generate sono nel formato IF_THEN. Ad esempio, se le persone acquistano un oggetto A, allora acquistano anche l'oggetto B. Affinché l'algoritmo possa derivare tali conclusioni, osserva il numero di persone che hanno acquistato l'oggetto B mentre acquistavano l'oggetto A. In questo modo viene calcolato un rapporto tra i due acquisti.

Principio base su cui funziona l'algoritmo Apriori:

- Se un set di elementi si verifica frequentemente, anche tutti i subset del set di elementi si verificano frequentemente.
- Se un set di elementi si verifica raramente, tutti i superset[8] del set di elementi hanno un evento non frequente.

Principali vantaggi dell'algoritmo Apriori:

1. È facile da implementare e può essere facilmente parallelizzato.
2. L'implementazione di Apriori si avvale di grandi proprietà di set di oggetti.

[8] Un set B è un subset di A, equivale a dire che A è un superset di B.

Principali applicazioni dell'algoritmo:

- *Rilevazione di reazioni avverse ai farmaci* - L'algoritmo viene utilizzato per l'analisi delle associazioni sui dati sanitari. Questa analisi produce delle regole di associazione che aiutano a identificare la combinazione di caratteristiche del paziente e farmaci che portano a effetti collaterali avversi dei farmaci.
- *Analisi del paniere di mercato* – Molte aziende nel commercio elettronico utilizzano Apriori per trarre informazioni sui dati su quali prodotti possono essere acquistati insieme e quali sono i più sensibili alla promozione.
- *Completamento automatico delle applicazioni* – Il completamento automatico di Google è un'altra applicazione popolare di Apriori in cui, quando l'utente digita una parola, il motore di ricerca cerca altre parole associate che le persone solitamente digitano dopo quella parola specifica.

2.3.5 Linear Regression Machine Learning Algorithm

L'algoritmo di regressione lineare mostra la relazione tra due variabili e il modo in cui il cambiamento di una variabile influisce sull'altra. L'algoritmo mostra l'impatto sulla variabile dipendente al cambiamento della variabile indipendente. Le variabili indipendenti sono indicate come variabili esplicative, in quanto spiegano i fattori che influenzano le variabili dipendenti. Le variabili dipendenti, invece, vengono spesso indicate come fattori di interesse o predittorie.

Principali vantaggi dell'algoritmo Regressione lineare:

1. È uno degli algoritmi di apprendimento automatico più interpretabili, facile da spiegare ad altre persone.
2. È facile da usare in quanto richiede una messa a punto minima.
3. È la tecnica di apprendimento automatico maggiormente utilizzata e quindi si sviluppa più velocemente.

Principali applicazioni dell'algoritmo:

- *Stima delle vendite:* La regressione lineare trova grande utilità per le previsioni di vendita basate sulle tendenze.
- *Valutazione del rischio:* L'algoritmo aiuta a valutare il rischio coinvolte nel settore assicurativo o finanziario. Tali risultati dell'analisi svolgono un ruolo vitale nelle decisioni aziendali importanti e vengono presi in considerazione per il reso conto del rischio.

Capitolo 3

Crowdsourcing

Il termine crowdsourcing fu utilizzato per la prima volta da Joff Howe in un articolo del 2006 per la rivista Wired, *The Rise of Crowdsourcing*[1]. In questo articolo, Howe, utilizza degli esempi per spiegare le potenzialità nell'effetturare richieste aperte ad un grande numero di persone per abbattere i costi e trarre il massimo risultato. Secondo Howe, la potenzialità del crowdsourcing si basa sul concetto che, siccome si tratta di una richiesta aperta a più persone, si potranno riunire quelle più adatte a svolgere determinate attività, a risolvere problemi di una certa complessità, e a contribuire con idee nuove e sempre più utili.

3.1 Che cosa è il crowdsourcing?

Il termine "Crowdsourcing" è un concetto relativamente recente che comprende molte pratiche e sfaccettature. A causa di quest'ultime, nel tempo, numerosi autori hanno voluto esprimere la propria definizione del termine crowdsourcing accompagnata da opportuni esempi per supportare la propria idea. Questo ha causato la creazione di molteplici definizioni tra cui alcune in conflitto con altre.
Estellés Arolas e Gonzàles Fernando pubblicarono nel 2012 un articolo, chiamato *Towards an integrated crowdsourcing definition*[2], in cui analizzarono le molteplici definizioni create traendo da ognuna i punti di forza e i punti comuni con le altre cercando di costruire una definizione che integri anche tutte le altre.

La definizione che diedero è la seguente:

> *"Il crowdsourcing è una tipologia di attività online partecipativa nella quale una persona, istituzione, organizzazione non a scopo di lucro o azienda propone ad un gruppo di individui, mediante un annuncio aperto e flessibile, la realizzazione libera e volontaria di un compito specifico. La realizzazione di tale compito, di complessità e modularità variabile, e nella quale il gruppo di riferimento deve partecipare apportando lavoro, denaro, conoscenze e/o esperienza, implica sempre un beneficio per ambe le parti. L'utente otterrà, a cambio della sua partecipazione, il soddisfacimento di una concreta necessità, economica, di riconoscimento sociale, di autostima, o di sviluppo di capacità personali, il crowdsourcer d'altro canto, otterrà e utilizzerà a proprio beneficio il contributo offerto dall'utente, la cui forma dipenderà dal tipo di attività realizzata."*

Da questa definizione è possibile estrarre otto caratteristiche che un'attività deve soddisfare in tutto o in buona parte per essere considerata un'attività di crowdsourcing.

Le otto caratteristiche di cui stiamo parlando sono le seguenti:

1. C'è una folla chiaramente definita.
2. Esiste un'attività con un obbiettivo chiaro.
3. Il risarcimento ricevuto dalla folla è chiaro.
4. Il crowdsourcer[9] è chiaramente identificato.
5. Il compenso che deve essere ricevuto dal crowdsourcer è chiaramente definito.
6. È un processo assegnato online di tipo partecipativo.
7. Utilizza una chiamata aperta di estensione variabile.
8. Usa Internet.

Nell'articolo citato si è cercato anche di analizzare le principali attività di crowdsourcing presenti fino a quel momento per verificare che rispettassero le otto caratteristiche della definizione. Le attività analizzate sono state: Wikipedia, InnoCentive, Threadless,

[9] Il crowdsourcer viene definito come colui che otterrà ed utilizzerà a proprio beneficio il contributo offerto dagli utenti.

Amazon Mechanical Turk, ModCloth, Youtube, Lànzanos, Delicious, Fiat Mio, iStockPhoto e Flickr.

La tabella sottostante mostra la valutazione delle otto caratteristiche per le singole iniziative, assegnando un "+" a una caratteristiche che appare chiaramente ed un "-" a quelle caratteristiche che non appaiono.

Tabella 1. Verifica della definizione. Fonte: articolo di Estellés e Gonzàles

	a	b	c	d	e	f	g	h
Wikipedia	+	+	+	-	-	+	-	+
InnoCentive	+	+	+	+	+	+	+	+
Threadless	+	+	+	+	+	+	+	+
Amazon Mechanical Turk	+	+	+	+	+	+	+	+
ModCloth	+	+	+	+	+	+	+	+
YouTube	+	-	-	-	-	-	-	+
Lánzanos	+	+	+	+	+	+	+	+
Delicious	+	-	-	-	-	-	-	+
Fiat Mio	+	+	+	+	+	+	+	+
iStockPhoto	+	+	+	+	+	+	+	+
Flickr	+	-	-	+	-	-	-	+

Secondo la tabella esistono alcuni casi chiari di crowdsourcing quali InnoCentive, ThreadLess, Amazon Mechanical Turk, Lànzanos, iStockPhoto, ModCloth e Fiat Mio. Dall'altra parte, invece, alcuni casi non sono identificati come crowdsourcing. Nel caso di Delicious, sei caratteristiche non sono identificate: un compito con un obbiettivo chiaro, il risarcimento dovuto dalla folla, il crowdsourcer e il vantaggio che riceve (la società non si comporta come un crowdsourcer e non riceve alcun beneficio dal lavoro della folla), la natura partecipativa del compito (non si può dire che sia un processo partecipativo in cui tutti gli utenti cercano lo stesso obiettivo finale) e l'esistenza di una chiamata aperta. Per questi motivi, si può affermare che Delicious non può essere considerato un esempio di crowdsourcing.

3.2 Crowdsourcing unito al Machine Learning

Viste le innumerevoli potenzialità offerte dal crowdsourcing combinate alle facilità di utilizzo, quest'ultimo è stato utilizzato anche nel campo del machine learning. Gli algoritmi di machine learning per un corretto funzionamento necessitano di dati di addestramento e il crowdsourcing è il modo perfetto per ottenere dati di alta qualità in base a ciò di cui l'algoritmo ha bisogno. Possiamo dire che il machine learning basato su crowdsourcing utilizza quest'ultimo per raccogliere una grande quantità di dati di formazione per l'algoritmo di apprendimento automatico.

In questo modo, il crowdsourcing ha introdotto un metodo per ottenere i dati, in particolare i dati etichettati, in modo veloce e a basso costo. Questa abbondanza di dati è stato un grande vantaggio per il machine learning, tuttavia, si è rilevato che questo metodo genera in maniera maggiore dati errati rispetto ai tradizionali metodi di annotazione. Questo lato negativo ha generato un interesse significativo nello sviluppo di meccanismi efficaci di controllo della qualità al fine di migliorare la qualità dei dati.

Questi due aspetti (controllo della qualità e machine learning) vengo analizzati in un articolo di Matthew Lease del 2011, *"On quality control and machine learning in crowdsourcing"*[3].

Secondo Lease, un controllo di qualità efficace svolge un ruolo importante nel determinare il successo di qualsiasi raccolta di dati tramite crowdsourcing. Nel suo articolo divide il controllo di qualità in diverse aree:

1. *Fattore umano* – Dal momento che sono le persone a formare la folla, il crowdsourcing può essere considerato un'impresa umana-centrica. Per questo motivo i fattori umani meritano una particolare considerazione per la progettazione e l'uso efficace del crowdsourcing.
2. *Automazione* – Creare dei sistemi di controllo di qualità automatici ed inserirli a livello di sistema di motori di raccolta dati. Questo permette di creare un utile livello di astrazione per i professionisti, che possono concentrarsi

sull'articolazione delle loro specifiche linee guida di annotazione e lasciare che il sistema si preoccupi del controllo di qualità a basso livello.

3. *Annotazione* – Creare delle linee guida di annotazione che coprano l'infinita varietà di dati riscontrati nella pratica e che siano internamente coerenti.
4. *Organizzazione dei lavoratori e dei compiti* – Assegnare dei ruoli e compiti largamente tradizionali ai lavoratori distribuiti in modo da ottimizzare le annotazioni uomo-macchina.
5. *Come distinguere le rare intuizioni dagli errori* – Introdurre nuovi modi per ascoltare le diverse opinioni dei contribuenti in modo da riuscire a distinguere quelle rare intuizioni rispetto alla massa generale.

Attingendo a questi punti sopra citati si possono sfruttare al massimo le potenzialità del crowdsourcing, ovvero, raccogliere enormi quantità di dati etichettati di alta qualità da usare come data set di addestramento per gli algoritmi di machine learning, i quali possono imparare da questi dati per perfezionare i propri modelli.

3.3 Esempi di sistemi sviluppati

Da quando si è capito come sfruttare il crowdsourcing per migliorare il machine learning sono stati sviluppati numerosi metodi per la classificazione di dati, prodotti, immagini, ecc.., che sfruttano le potenzialità di questo "nuovo" metodo per la raccolta di grandi quantità di dati a basso costo. Di seguito illustrerò una breve descrizione dei lavori più rilevanti nell'ambito della classificazione di oggetti.

3.3.1 Chimera: Large-Scale Classification using Machine Learning, Rules, and Crowdsourcing

Chimera[4] è un sistema di classificazione sviluppato da Chong Sun, Narashiman Rampalli, Frank Yang ed AnHai Doan, che impiega più classificatori al suo interno, basati sia sull'apprendimento che su regole. Il sistema impiega il crowdsourcing per

valutare il risultato della classificazione, segnalare casi problematici e quindi inoltrare tali casi agli analisti interni. Quest'ultimi analizzano i casi, scrivono regole per affrontarli, correggono le etichette ed incorporano il feedback nel sistema così da consentire un miglioramento continuo nel tempo.

Chimera è formato da 4 aspetti fondamentali:

- Utilizza sia le regole di apprendimento sia quelle elaborate a mano (scritte dagli analisti di dominio).
- Utilizza una combinazione di analisti interni e crowdsourcing per valutare ed analizzare il sistema, per raggiungere una soluzione sempre più accurata e in continuo miglioramento.
- Chimera è scalabile in termini di risorse umane, utilizzando analisti interni e sfruttando il crowdsourcing.
- Chimera usa un algoritmo ibrido uomo-macchina che tratta l'apprendimento, le regole, i lavoratori della folla, gli analisti interni e gli sviluppatori come "cittadini di prima classe".

In breve, Chimera funziona nel seguente modo:

1. Inizializza il sistema utilizzando un set base di dati di allenamento e regole fatte a mano fornite dagli analisti. Forma i classificatori basati sull'apprendimento.
2. Loop:
    a. Dato un insieme di articoli in arrivo, li classifica e utilizza il crowdsourcing per valutare continuamente i risultati e i casi di etichettatura giudicati errati dalla folla.
    b. Gli analisti esaminano i casi segnalati e li correggono scrivendo nuove regole, rietichettando gli oggetti e avvisando gli sviluppatori.
    c. Le regole appena create e gli articoli reclamati sono incorporati nel sistema e gli sviluppatori possono perfezionare l'algoritmo automatico sottostante.
    d. Gli articoli che il sistema rifiuta di classificare vengono esaminati dagli analisti che creano regole fatte a mano e dati di addestramento per permette al sistema di riuscire a classificare questi articoli in futuro. Le

regole e i dati di addestramento appena creati sono incorporati nel sistema e viene eseguito di nuovo il primo passo.

3.3.2 Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets

Revolt[5] è un sistema di crowdsourcing collaborativo per la generazione di dataset di dati etichettati per il machine learning sviluppato da Joseeph Chee Chang, Saleema Amershi e Ece Kamar. Questo sistema divide un set di dati in più lotti e quindi coordina i lavoratori (crowdworker) per creare etichette per determinati oggetti (oggetti che ricevono etichette unanime da più crowdworkers) in ogni lotto e identificare gli elementi incerti (oggetti che ricevono etichette in conflitto) per ulteriori spiegazioni ed elaborazioni. Revolt possiede due versioni: una sincronizzata ed un asincrona. Nella versione sincronizzata, il sistema coordina piccole squadre di tre crowdworkers attraverso tre fasi sincronizzate: vota, spiega e poi classifica. Nella versione asincrona, invece, il sistema sollecita diversi crowdworkers a lavorare in modo indipendente nelle fasi di votazione ed esplicazione, mantenendo lo stesso giudizio di tre persone per oggetto eliminando, quindi, il costo di coordinamento di tre operatori sincronizzati. Dopo aver raccolto i giudizi e le spiegazioni della folla in tutti i gruppi, entrambi i sistemi producono algoritmicamente strutture a vari livelli per la revisione da parte di chi ha richiesto la classificazione, per determinare l'etichettatura finale.

Il sistema, quindi, è composta da tre fasi principali (versione sincronizzata):

1. *Fase di votazione* – Revolt, inizialmente, mantiene i lavoratori dentro ad una "stanza" fino a quando non accettano il compito e possono iniziare come gruppo. La fase del voto inizia, quindi, raccogliendo giudizi di etichette indipendenti da più operatori usando un'interfaccia simile ad una tradizionale etichettatura con crowdsourcing. Il sistema oltre a mostrare le opzioni predefinite di etichettatura, include anche un'opzione "forse/non sicuro" per garantire che i lavoratori della folla non siano obbligati a prendere decisioni arbitrarie per incertezza. I crowdworker in questa fase sono informati che anche altri membri dello stesso

gruppo stanno etichettando gli stessi oggetti nello stesso istante e che gli sarà chiesto, nelle fasi successive, di confrontarsi a vicende sulle etichettature proposte. Prima che Revolt possa procedere alla fase successiva, tutti i lavoratori del gruppo devono finire di etichettare tutti gli oggetti a loro assegnati. I lavoratori che finiscono in anticipo vengono messi in una sala d'attesa dove possono vedere in tempo reale quanti del loro gruppo stanno ancora etichettando gli oggetti. Una volta ricevute tutte le etichettature, a determinati articoli vengono assegnate le etichette finali, mentre gli articoli "incerti" procedono alla fase esplicativa.

2. *Fase esplicativa* – In questa fase viene chiesto ai crowdworker di fornire una breve spiegazione sugli oggetti etichettati come "incerti" nella fase precedente. Ciascun lavoratore, attraverso delle istruzioni, è già a conoscenza che qualcuno del gruppo non è d'accordo su alcune etichette assegnate ad alcuni oggetti e, quindi, il suo compito è quello di descrivere la logica utilizzata per etichettare questi oggetti agli altri membri del gruppo.
3. *Fase di categorizzazione* – Il sistema, in questa fase, ripropone la categorizzazione degli elementi incerti ai crowdworker, con la differenza che adesso voteranno dopo aver sentito la spiegazione degli altri membri del gruppo di lavoro. Le categorie possono essere selezionate da un elenco, messo a disposizione per ciascun oggetto, di categorie esistenti oppure aggiunte manualmente tramite campo di inserimento testo. Ogni volta che un crowdworker aggiunge una nuova categoria, il sistema aggiorna dinamicamente la lista di quest'ultime. Il sistema, inoltre, è dotato di due meccanismi per incentivare l'utilizzo di categorie già esistenti. Il primo consiste nel suggerire le categorie esistenti mentre un lavoratore digita il nome di una nuova categoria nel campo testo dedicato. Il secondo, invece, consiste nell'ordinare la lista delle categorie in base al numero di crowdworker che hanno l'hanno utilizzata.

Dopo che tutti i crowdworkers di tutti i gruppi hanno superato tutte e tre le fasi, Revolt raccoglie i loro feedback e assegna, per alcuni oggetti, direttamente un'etichetta. Dopodiché crea strutture contenenti gli oggetti incerti applicando una maggioranza semplice per l'assegnazione dell'etichetta tra quelle proposte dai lavoratori. Nel caso in

cui tutti i crowdworkers suggeriscano etichette diverse, viene utilizzata un'etichetta casuale. Concluso il processo, le strutture possono essere presentate a chi ha richiesto le etichettature per la revisione e decisione finale.

Nella versione asincrona di Revolt alcune fasi cambiano leggermente:
Il gruppo di lavoratori (composto sempre da tre persone), nella fase di votazione, etichetta gli oggetti in modo indipendente. Nella fase esplicativa, gli elementi incerti vengono messi di nuovo sul "mercato" e fatti spiegare ad un gruppo diverso di crowdworker. Nello specifico, ad ogni crowdworker viene assegnato un oggetto ed un'etichetta proposta da un altro lavoratore e gli viene chiesto di giustificare quell'etichetta con la consapevolezza che lì ci sono state discrepanze nei voti. La versione asincrona di Revolt non prevede una fase di categorizzazione, ma utilizza le spiegazioni raccolte nella fase esplicativa per creare direttamente le strutture.

Capitolo 4

Sistema di Crowdsourcing sviluppato

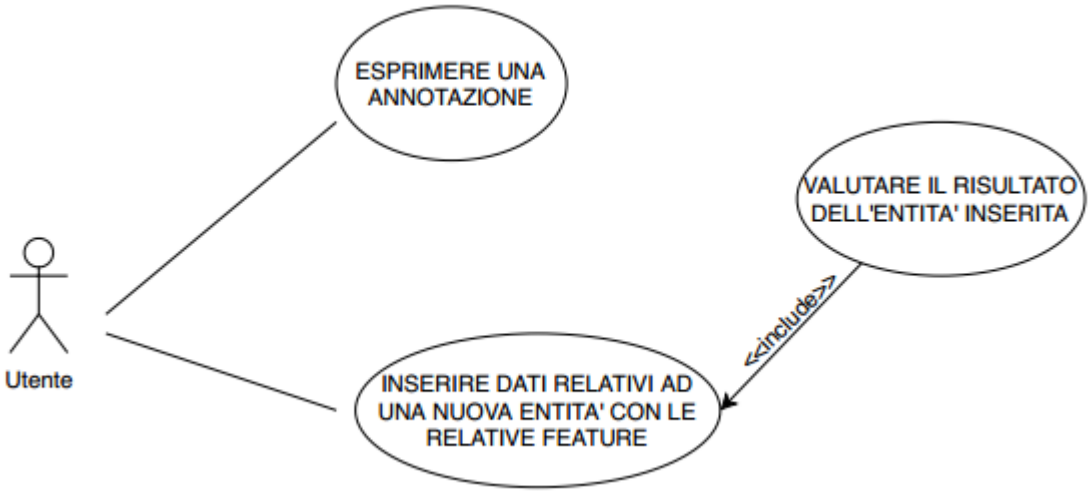
Il sistema sviluppato prende in input un dataset di dati classificati e sottopone queste classificazioni alla valutazione da parte degli utenti. Ogni valutazione viene salvata in un database così da renderle disponibili agli analisti per eventuali correzioni o miglioramenti dell'algoritmo di classificazione. Oltre alla valutazione di dati già classificati, il sistema offre la possibilità di "sfidare" l'algoritmo a classificare una nuova entità i cui dati sono inseriti dall'utente stesso. Dopo aver inserito quest'ultimi il sistema proporrà all'utente una classificazione che verrà valutata ed inserita nel database per una successiva analisi.

4.1 Analisi dei casi d'uso

L'analisi dei casi d'uso è una tecnica utilizzata per identificare i requisiti di un sistema e le informazioni utilizzate per definire i processi e le classi che verranno utilizzati nel diagramma dei casi d'uso. Questa analisi è la base su cui verrà costruito il sistema. Gli obbiettivi principali di questa analisi sono: progettare un sistema dal punto di vista dell'utente, comunicare il comportamento del sistema nei termini dell'utente e specificare tutti i comportamenti visibili esternamente. Il caso d'uso, da cui è formata questa analisi, è una tecnica usata per effettuare in maniera esaustiva e non ambigua, la raccolta dei requisiti al fine di produrre software di qualità. L'insieme dei casi d'uso

individua e descrive gli scenari elementari di utilizzo del sistema da parte degli attori che si interfacciano con esso.

Tabella 2. Diagramma dei casi d'uso



La tabella 3 individua gli scenari che il sistema permette all'utente di svolgere. Nel nostro caso le funzioni sono due: la prima è quella di esprimere una valutazione su una classificazione già effettuata; la seconda è quella di sfidare l'algoritmo a classificare una propria entità inserendo dei dati che la descrivono. Ci sarebbe anche una terza funzione, ovvero, quella di esprimere una valutazione sulla classificazione proposta dal sistema per l'entità inserita dall'utente. Quest'ultima funzione, però, è inclusa nella seconda in quanto l'utente non può esprimere una valutazione su un'entità di cui non ha inserito i dati.

Descrizione caso d'uso: Esprimere una annotazione

Attori: Utente utilizzatore del sistema

Pre-condizioni: Nessuna

Sequenza degli eventi:

1. *L'utente inserisce un nickname per essere identificato nella sessione;*
2. *Il sistema propone randomicamente un'entità da valutare tra quelle salvate;*
3. *L'utente esprime la sua valutazione attraverso gli appositi pulsanti;*
4. *L'utente invia la sua valutazione al sistema;*

5. *Il sistema notifica all'utente di aver ricevuto la valutazione correttamente.*

Sequenze alternative:

5. *Il sistema notifica all'utente di non aver ricevuto la valutazione correttamente.*

Descrizione caso d'uso: Inserire dati relativi ad una nuova entità con le relative feature

Attori: Utente utilizzatore del sistema

Pre-condizioni: Nessuna

Sequenza degli eventi:

1. *L'utente inserisce un nickname per essere identificato nella sessione;*
2. *L'utente accede alla scheda dedicata alla compilazione dei campi;*
3. *L'utente compila i campi necessari per inserire una nuova entità;*
4. *L'utente invia i dati compilati al sistema;*
5. *Il sistema restituisce una classificazione per l'entità inserita;*
6. *L'utente esprime una valutazione sulla classificazione ricevuta;*
7. *L'utente invia i risultati della valutazione;*
8. *Il sistema notifica l'utente che la sua entità è stata inserita correttamente.*

Sequenze alternative:

5. *Il sistema restituisce un errore verificatosi nella classificazione;*
6. *Il sistema non inserisce l'entità compilata dall'utente.*

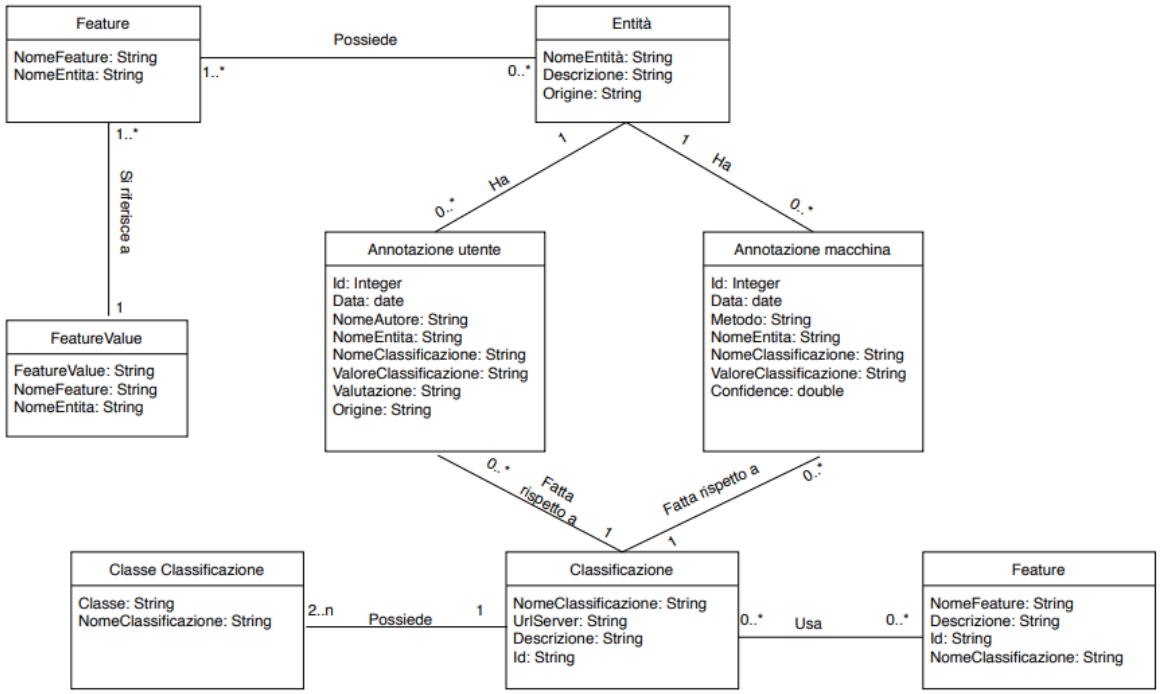
---

7. *L'utente non invia i risultati della valutazione;*
8. *Il sistema non inserisce l'entità dell'utente.*

4.2 Analisi di dominio

L'analisi del dominio è una delle attività che costituiscono l'analisi[10] e concorrono alla definizione delle specifiche di un sistema o applicazione software. Lo scopo di questa analisi è quello di comprendere a fondo i concetti, le dinamiche e le regole generali che definiscono il dominio applicativo[11] in cui il sistema software dovrà essere impiegato. La tabella sottostante mostra il diagramma di dominio finale del sistema sviluppato con tutte le tabelle con i relativi campi e le relazioni fra di esse.

Tabella 3. Diagramma di dominio del sistema



Dalla tabella si può notare come il sistema creato sia composto da otto tabelle:

- *Entità* – Questa tabella contiene tutti gli oggetti, chiamati entità, soggetti a valutazione da parte della folla. Gli oggetti possono derivare dal dataset iniziale di addestramento oppure dalle entità inserite dagli utenti attraverso la funzione offerta dal sistema.

[10] L'analisi ha lo scopo di chiarire, dettagliare e documentare le funzioni, i servizi e le prestazioni che devono essere offerti da un sistema software o programma.
[11] Per dominio applicativo ci si riferisce al contesto in cui un'applicazione software opera.

- *Feature* – Questa tabella contiene l'insieme dei nomi delle feature[12] di ogni entità salvata nel sistema.
- *Feature Value* – Contiene i valori delle feature salvate di ogni entità.
- *Annotazione Utente* – In questa tabella vengono salvate tutte le valutazioni, per ogni classificazione, effettuate dagli utenti.
- *Annotazione Macchina* – Questa tabella, invece, contiene per ogni entità, tutte le classificazioni effettuate dall'algoritmo.
- *Classificazione* – La tabella contiene il nome e le informazioni delle classificazioni gestite dal sistema, ovvero quelle che gli utenti devono valutare.
- *Classe Classificazione* – In questa tabella vengono salvate le possibili classi, per ogni classificazione, che l'algoritmo può assegnare alle varie entità.
- *Feature* – Contiene l'insieme delle feature di ogni classificazione, che ogni entità dovrebbe avere per essere classificata al meglio.

4.3 Architettura del sistema sviluppato

Il sistema è composto da 3 componenti principali, senza le quali non funzionerebbe nel modo corretto. Queste tre componenti sono: il database, il server di classificazione e l'interfaccia grafica.

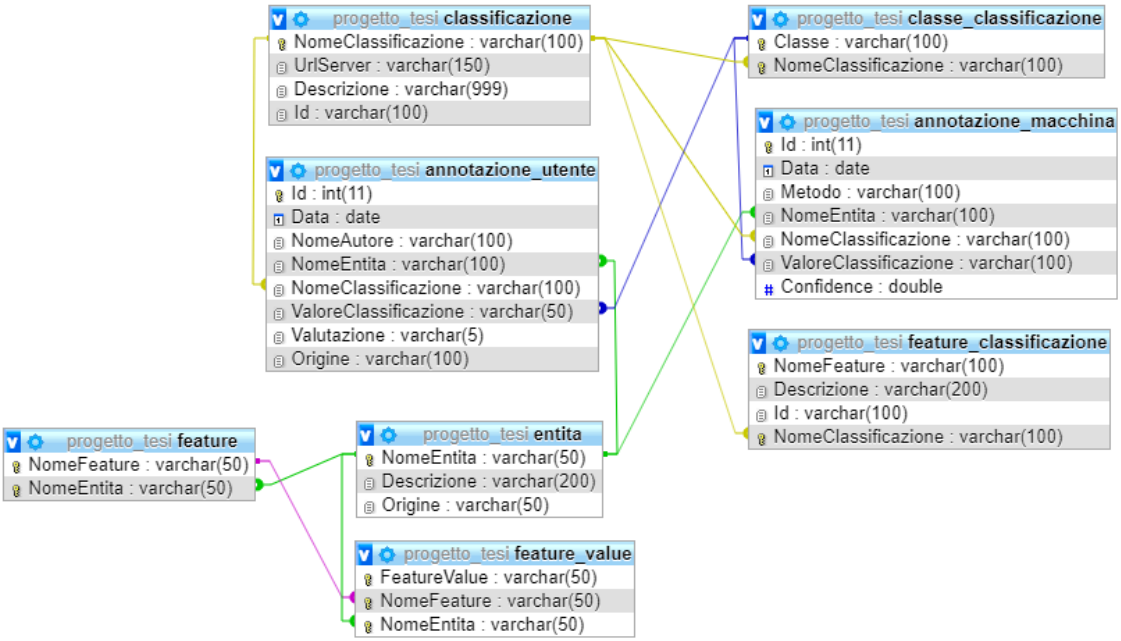
4.3.1 Database

Il database è una componente essenziale del sistema in quanto al suo interno vengono salvate tutte le informazioni riguardanti le entità, le annotazioni e le classificazioni. Il database finale del sistema è composto da otto tabella interconnesse attraverso le chiavi esterne. Ogni volta che un utente esprime una valutazione o inserisce una nuova entità, viene effettuata una chiamata INSERT al database per inserire i nuovi dati all'interno delle opportune taballe. Il database viene utilizzato anche in fase iniziale

[12] Per feature si intende una caratteristica dell'entità

per il caricamento delle entità precedentemente salvate in modo da proporre all'utente della sessione molteplici entità da valutare.

Figura 1. Struttura finale del database



4.3.2 Server adibiti alla classificazione

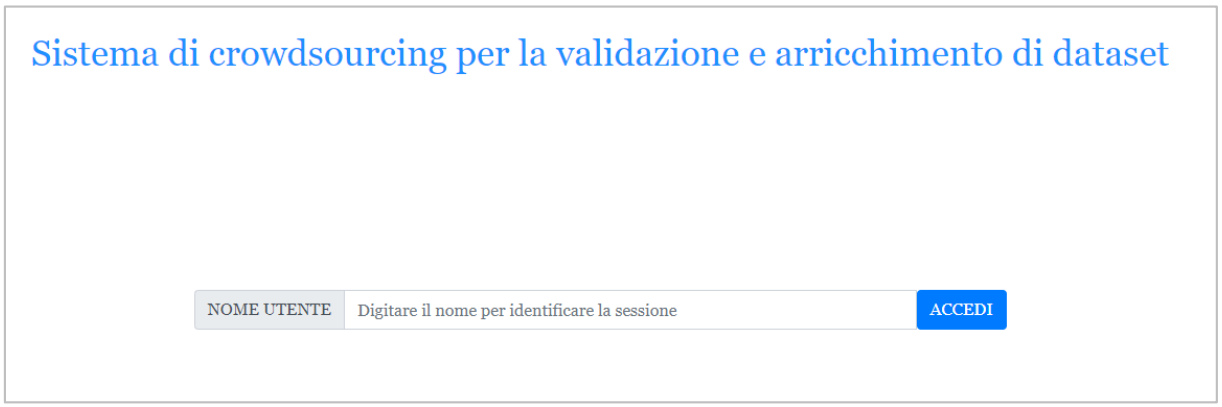
Il sistema per classificare le nuove entità inserita dagli utenti, effettua delle chiamate a server esterni adibiti appositamente per la classificazione di tali entità. I server prendono in input un file .json, effettuano la classificazione e restituiscono il risultato sempre attraverso un file .json. Nel caso in cui i server non riescano ad esprimere un risultato per l'entità inserita dall'utente, il risultato sarà una classificazione random tra le possibili classi della classificazione in questione. L'indirizzo per raggiungere ogni server è salvato nel database, per cui ogni qualvolta un utente invia una nuova entità, il sistema estrapola l'indirizzo del server per ogni classificazione gestita ed effettua le chiamate al relativo indirizzo. Se la chiamata al server fallisce, il sistema provvederà a restituire un errore all'utente.

4.3.3 Interfaccia grafica

Gli utenti hanno la possibilità di eseguire le funzioni messe a disposizione dal sistema attraverso l'interfaccia grafica. Nel processo di sviluppo si è dato maggiore priorità alla realizzazione delle funzioni essenziali a discapito della grafica visiva. Il risultato finale è stata un'interfaccia utente non troppo elaborata ma che permettesse di eseguire tutte le operazioni possibili. Quest'ultima è composta da tre pagine principali: una pagina dedicata all'autenticazione, una per le due funzioni principali ed una pagina dedicata all'amministratore.

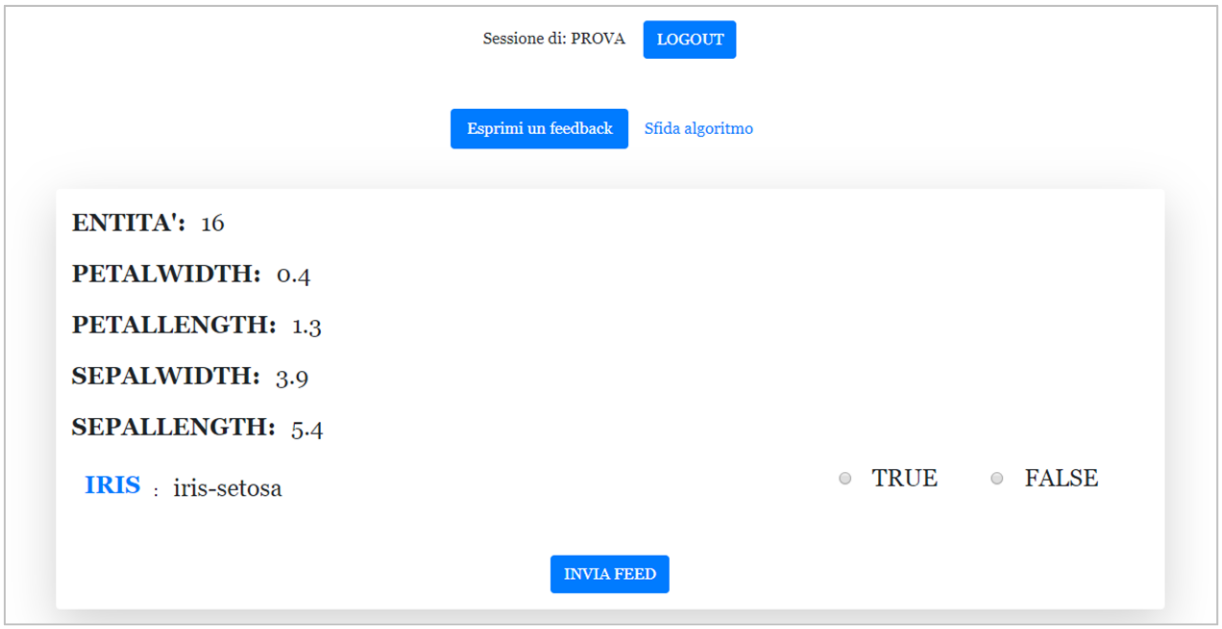
La pagina di autenticazione, nonché quella iniziale, è composta da un campo testo dove inserire un proprio nickname per essere identificati nella sessione. Lo scopo di questo nickname è far in modo di proporre all'utente solo le entità che non ha valutato nella sessione in corso.

Figura 2. Schermata iniziale del sistema.



La pagina principale, a cui si arriva dopo aver inserito un nickname, contiene un menu formato da due pulsanti, uno per accedere all'area dedicata alla valutazione di entità già classificate, ed uno per accedere all'area dedicata all'inserimento di una nuova entità da far classificare. Nella sezione dedicata alla prima funzione (valutazione di entità già classificate) vengono visualizzate le informazioni di un'entità casuale tra quelle salvate nel database con la classificazione proposta ed i relativi pulsanti per esprimere la propria opinione (vero o falso).

Figura 3. Sezione dedicata alla prima funzione con dataset di prova Iris.



Nella seconda sezione, invece, vi sono i campi di input in cui l'utente deve inserire i dati dell'entità che vuole registrare e far classificare al sistema.

Figura 4. Sezione dedicata all'inserimento di una nuova entità

Infine, vi è una pagina a cui solo l'amministratore del sistema può accedere per svolgere funzioni a lui dedicate. In questa sezione, l'amministratore è in grado di caricare dati di addestramento, scaricare i dati relativi ai feedback ricevuti dagli utenti e le entità inserite da quest'ultimi.

Figura 5. Pagina dedicata all'amministratore del sistema

4.4 Meccanismi e funzionamento del sistema

In questo paragrafo verrà analizzato come lavora il sistema e quali meccanismi scaturiscono durante le principali funzioni, ovvero, la valutazione di un'entità, l'inserimento di una nuova entità e le operazioni dedicate solo ed esclusivamente all'amministratore.

4.4.1 Valutazione di entità già classificate

Il sistema, per proporre le entità già classificate da far valutare ai crowdworkers, effettua una chiamata al database per recuperare tutte le entità salvate, le feature collegate a quest'ultime e le relative classificazioni (annotazioni). Una volta completato ciò, salva tutte le entità dentro ad una variabile di sessione collegata al nickname inserito

dall'utente nella schermata iniziale. L'entità che viene proposta ogni volta è estratta casualmente dalla lista totale ed ogni volta che l'utente completa con successo la votazione, l'entità votata viene rimossa dalla suddetta lista in modo da non riproporla all'utente della sessione. Questo meccanismo fa sì che un'entità venga votata al massimo una volta per ogni sessione. La votazione consiste nell'esprimere una scelta "vero" o "falso" per ogni classificazione proposta dell'entità estratta. Nel caso in cui l'utente completi la procedura con successo, il sistema provvederà ad effettuare una chiamata al database per salvare la votazione e riproporrà al lavoratore un'altra entità secondo la procedura descritta precedentemente. Nel caso in cui l'utente effettui la disconnessione dalla sessione in corso, il meccanismo sopra descritto verrà azzerato con la possibilità che venga riproposta un'entità già valutata in una sessione precedente.

4.4.2 Inserimento di una nuova entità

Per l'inserimento di una nuova entità, il sistema richiede all'utente la compilazione di alcuni campi fondamentali per una corretta classificazione da parte dell'algoritmo. Questi campi sono: nome dell'entità che si vuole inserire, una descrizione delle caratteristiche principali ed un insieme di campi dinamici (numero di campi a discrezione dell'utente) rappresentanti le features[13] e i valori di queste. Una volta compilato ed inviato il tutto, il sistema provvederà a raccogliere i dati inseriti e a inserirli all'interno di un file json. Questo file verrà inviato ai server di ogni classificazione, i quali utilizzeranno le informazioni scritte all'interno per restituire una loro classificazione dell'entità. Le classificazioni inviate dai server saranno proposte all'utente, il quale dovrà esprimere un giudizio (sempre "vero" o "falso") sulla correttezza o meno di quest'ultime in base a ciò che rappresenta l'entità che si voleva inserire. Il sistema, infine, raccoglierà i giudizi dell'utente ed effettuerà una chiamata al database per salvare: l'entità inserita dall'utente, la classificazione proposta dagli algoritmi e le valutazioni fatte su di esse.

[13] Per feature si intendono le caratteristiche dell'oggetto. (es. "la mela è di colore rosso", colore è il nome della feature mentre rosso è il valore della feature.

Casi speciali:

- Se un utente non esprime una valutazione sulle classificazioni proposte, il sistema non salverà l'entità inserita ma annullerà tutta la procedura.
- Se tutte le feature inserite dall'utente non sono utilizzabili dall'algoritmo per una corretta classificazione, quest'ultimo restituirà come risultato una classificazione casuale tra quelle disponibili e la proporrà all'utente.
- Se un server adibito alla classificazione non risulta raggiungibile, il sistema provvederà ad annullare l'operazione e a notificare l'utente con un messaggio di errore.

4.4.3 Operazioni dedicate all'amministratore

Nella pagina iniziale, se viene inserito il nickname "admin" comparirà un ulteriore campo di testo dove l'amministratore potrà inserire la propria password per accedere alla pagina a lui dedicata. In questa pagina potrà eseguire tre operazioni, ovvero, caricare un dataset di entità già classificate, scaricare un file contenente tutti i feedback lasciati dai crowdworker e scaricare un file contenente tutte le entità inserite dagli utenti con le relative classificazioni e feedback su di esse.

Caricamento di un dataset di entità già classificate:

Per svolgere questa operazione l'amministratore dovrà caricare, attraverso l'apposito pulsante, un dataset contenuto in un file di tipo json. Il sistema è stato sviluppato per gestire solo questo formato, per cui l'inserimento di un qualsiasi altro tipo di file restituirà un errore. Nel caso in cui il nome del file sia già presente negli archivi del sistema, l'operazione verrà annullata e verrà notificato all'amministratore che il file è già presente. Se il file caricato non rientrerà tra i casi sopra descritti, il sistema provvederà ad analizzarlo estraendo le informazioni delle entità con le relative classificazioni. Una volta estratti tutti i dati, verrà effettuata una chiamata al database

per il salvataggio di questi e terminata questa operazione verrà inviata una notifica all'amministratore di avvenuto caricamento del dataset selezionato.

Download dei feedback rilasciati dagli utenti:

Tramite il pulsante nominato "Download users feedback" è possibile scaricare un file json contenente tutti i feedback degli utenti. Quando il sistema riceve questa richiesta, effettua una chiamata al database per recuperare tutte le informazioni necessarie. Nel database non vengono salvati i conteggi dei feedback per ogni classe, per cui sarà il sistema che dovrà organizzare le informazioni ricevute ed effettuare un conteggio dei feedback ottenuti per ogni classe. Per esempio, se l'oggetto mela può essere classificato come "oggetto fisico" (classe della classificazione "x"), il sistema dovrà contare quanti utenti hanno votato true per "oggetto fisico". Una volta effettuati tutti i conteggi, questi verranno assegnati all'entità a cui fanno riferimento ed inseriti in un file json che verrà fatto scaricare.

Download delle entità inserite dagli utenti:

Tramite il pulsante "Download users entity", invece, è possibile scaricare sempre un file json in cui sono contenute tutte le entità, e le informazioni collegate, inserite dagli utenti. Per ottenere ciò verrà effettuata una chiamata al database per ottenere le informazioni necessarie. Queste informazioni saranno riorganizzate tramite appositi meccanismi ed inserite nel file che il sistema farà scaricare all'amministratore.

Capitolo 5

Conclusioni

Il presente lavoro ha cercato di rispondere alla domanda: *"È possibile costruire un sistema che permetta in maniera semplice di validare annotazioni di un dataset ed estendere il dataset con nuove annotazioni e nuove entità?"*.

La risposta a questa domanda è positiva in quanto il software creato soddisfa pienamente i requisiti della domanda e ci permette di affermare che l'obbiettivo della tesi è stato raggiunto.

Tuttavia, la creazione di questo sistema ha fatto emergere alcune criticità per quanto riguarda la scalabilità del progetto. Nello specifico, il software è stato sviluppato e testato facendo riferimento al dataset iniziale "IRIS" contenente delle classificazioni di piante. Durante lo sviluppo si è cercato di generalizzare quante più procedure possibili, ma alcuni elementi, purtroppo, sono legati al dataset di addestramento utilizzato. Uno di questi elementi è l'estrazione dei dati da salvare nel database contenuti nel dataset inserito dall'amministratore. La procedura è sì generalizzata, ma legata alla struttura del dataset "IRIS", ovvero, qualunque dataset scritto nello stesso formato di IRIS verrà estratto in modo corretto, mentre nel caso in cui il formato sia diverso, non è garantita la corretta esecuzione della procedura. In questo caso, dunque, saranno necessarie delle piccole modifiche per adattare la procedura di estrazione dei dati al formato di dataset che si vuole utilizzare.

Lo sviluppo di questo sistema è stato pensato per raggiungere gli obbiettivi base predisposti all'inizio della progettazione per cui vi sono molte estensioni che si potrebbero aggiungere in un futuro.
Per esempio si potrebbe aggiungere:

- Un interfaccia grafica più completa in modo da invogliare i crowdworker a lavorare con maggiore serietà e concentrazione.
- Introdurre più tipi di classificazione in quanto il sistema attuale è in grado di gestire solo delle classificazioni multiclasse a singola label, ovvero permette la verifica dei dati associati a delle classificazioni multiclasse a singola label. Sarebbe utile ampliare questo scenario con le altri tipi di classificazioni come la multiclasse multi label, ecc…
- Introdurre una fase esplicativa da proporre all'utente dopo che quest'ultimo ha espresso una votazione su una classificazione, in modo che gli esperti in fase di revisione abbiano una spiegazione logica del voto inserito.
- Introdurre procedure che riconoscano la maggior parte dei formati di dataset in circolazione così da limitare le modifiche per l'adattamento del software ai dataset.

Queste estensioni non sono state sviluppato a causa del limitato tempo di progettazione ed esecuzione, ma che in un futuro potrebbero essere eseguite per portare il software ad un livello maggiore di completezza e scalabilità.

Bibliografia

[1] Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, *14*(6), 1-4.

[2] Estellés Arolas, E.; González Ladrón-de-Guevara, F. (2012) Towards an integrated Crowdsourcing definition. Journal of Information Science. Vol 38. no 2. 189-200.

[3] Lease, M. (2011, August). On quality control and machine learning in crowdsourcing. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[4] Sun, C., Rampalli, N., Yang, F., & Doan, A. (2014). Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. *Proceedings of the VLDB Endowment*, *7*(13), 1529-1540.

[5] Chang, J. C., Amershi, S., & Kamar, E. Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets.

Sitografia

Agenda digitale, https://www.agendadigitale.eu/cultura-digitale/linguaggio-naturale-e-intelligenza-artificiale-a-che-punto-siamo/

Awhy, https://www.awhy.it/blog/come-funziona-lintelligenza-artificiale/

Dezyre, https://www.dezyre.com/article/top-10-machine-learning-algorithms/202

Intelligenza artificiale, http://www.intelligenzaartificiale.it/intelligenza-artificiale-forte-e-debole/

Intelligenza artificiale, http://www.intelligenzaartificiale.it/machine-learning/

Leganerd, https://leganerd.com/2015/07/17/breve-storia-dellintelligenza-artificiale/

Makeuseof, https://www.makeuseof.com/tag/machine-learning-algorithms/

Quora, https://www.quora.com/What-is-crowdsourced-machine-learning

Sas, https://www.sas.com/it_it/insights/analytics/machine-learning.html

Techopedia, https://www.techopedia.com/definition/190/artificial-intelligence-ai

Wikipedia, https://it.wikipedia.org/wiki/Intelligenza_artificiale

Wikipedia, https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning

Wikipedia, https://it.wikipedia.org/wiki/Teorema_di_Bayes

Wikipedia, https://it.wikipedia.org/wiki/Classificatore_bayesiano

Wikipedia, https://it.wikipedia.org/wiki/Apprendimento_automatico

Wikipedia, https://it.wikipedia.org/wiki/Crowdsourcing

Wikipedia, https://it.wikipedia.org/wiki/Analisi_del_dominio

Wikipedia, https://en.wikipedia.org/wiki/Use-case_analysis

Wikipedia, https://it.wikipedia.org/wiki/Caso_d%27uso_(informatica)