UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

DIPARTIMENTO DI INFORMATICA

Corso di Laurea in

Sicurezza dei Sistemi e delle Reti Informatiche

(L-31 Scienze e Tecnologie Informatiche)

Studio e sviluppo di connettori di scraping di sorgenti social network per ingestione in sistemi di Big Data Analytics

Relatore:

Prof. Marco Anisetti

Tesi di Laurea di:

Francesco Avantaggiato

Matr. 942790

ANNO ACCADEMICO 2021/2022

A mio Fratello

Indice

Capitolo 1

Introduzione

1.1 Il progetto

Il progetto sviluppato per questo lavoro di tesi ha riguardato l'attività del web scraping da fonti social network e la sua applicazione nella programmazione di connettori di dati in sistemi di Big Data Analytics. Lo studio di soluzioni valide ha individuato come obiettivi principali i social di Meta Inc., ovvero Facebook ed Instagram, per la loro vasta utenza e conseguente ingente quantità di dati pubblici.

Nella tesi vengono affrontate tematiche inerenti alla legalità delle azioni di scraping, in quanto ad oggi, l'attività ricade in una "zona grigia", non essendo previste direttamente norme. I social tramite i loro termini di servizio, vietano qualsiasi tipologia di estrazione dati, anche adottando soluzioni tecniche, per salvaguardare le informazioni pubblicate dai propri utenti.

Oggetto di attenzione all'interno del progetto è anche l'impiego dei dati estratti, presentando valide proposte d'utilizzo per scopi di investigazione e Open Source Intelligence.

Lo sforzo tecnico e progettuale è consistito principalmente nella creazione di valide metodologie di elusione dei controlli anti-scraping attuati dalle piattaforme. Lo studio del contrasto ha garantito la progettazione di sistemi attuabili in entrambi i connettori dei social target, uniformando la loro gestione e il loro output.

1.2 Organizzazione dei contenuti

Il presente lavoro di tesi è così strutturato:
Il Capitolo 2 presenta lo stato dell'arte rispetto al web scraping in generale e da fonti social network. Vengono inoltre citati gli scenari d'impiego dei dati in ambito investigativo e gli aspetti giuridici dell'attività.
Il Capitolo 3 individua i requisiti di un progetto di web scraping presentando le tecnologie di base da impiegare, le soluzioni ideali di elusione dei controlli anti-scraping e la gestione dei dati in output.
Nel Capitolo 4 viene presentato il lavoro di studio e sviluppo svolto per il progetto di tesi. In particolare per i singoli connettori di Facebook ed Instagram vengono proposti i punti salienti del progetto e le soluzioni implementative attuate. Inoltre vengono elencate ed esplicate tutte le tecnologie di sviluppo adottate.
Il Capitolo 5 presenta i test effettuati sul software, proponendo un confronto sia sulle versioni dei tool open source impiegati che sull'efficienza in termini di dati, tempo e resistenza dei connettori sviluppati.
Nel Capitolo 6 vengono esposte le conclusioni sul progetto sviluppato, presentando eventuali sviluppi futuri.
Infine, nell'Appendice, A sono presentate parti del codice prodotto ed esempi di output dei connettori.

Capitolo 2

Stato dell'arte

Le applicazioni di estrazione dati nel web aumentano esponenzialmente con il valore delle informazioni. La vasta eterogeneità consente lo sviluppo e lo svolgimento di quest'attività in ambiti differenti.

Ad oggi, porre come obiettivo di scraping un social network vuol dire catalogare e gestire enormi quantità di dati. La presenza di interesse delle società e delle community open source al tema dello scraping dei social, individua l'importanza delle informazioni memorizzate su questi servizi e la conseguente protezione degli stessi da parte dei gestori delle piattaforme.

Con la presenza di regolamentazioni e leggi a carattere statale ed internazionale si nota come i gestori dei servizi siano stati nel tempo obbligati a cambiare l'approccio nei confronti della gestione dei dati pubblici. Inoltre, la dichiarazione di particolari termini d'uso delle piattaforme, garantisce un'apparente protezione dell'utente nei confronti dell'utilizzo dei dati per scopi non previsti.

Nel seguente capitolo si presenterà lo stato dell'arte del web scraping, con particolare attenzione ai social network.

2.1 Web Scraping

Con il termine "web scraping" si fa riferimento all'attività di raccolta automatica di informazioni da Internet inglobando sia varie tecniche di programmazione che diverse tecnologie web. Solitamente l'acquisizione di dati da un servizio avviene attraverso le Application Programming Interface (paragrafo 3.1.4) messe a disposizione dallo stesso, in grado di fornire output formattati e standardizzati (es. JSON), pronti ad essere analizzati.[1]

Nel campo della ricerca ed estrazione dei dati su internet, è necessario prendere in considerazione anche l'esistenza dell'attività nota come "web crawling". Il web crawling, a differenza dello scraping, viene impiegato per l'analisi e l'indicizzazione dei siti web ed utilizzato da parte dei motori di ricerca per fornire i risultati basati sulle parole chiave richieste dagli utenti. L'applicazione in grado di effettuare l'attività di crawling è definita "web crawler", nota anche come "spider" o "robot".[2]

L'ingente quantità di dati eterogenei presenti in Internet implica un maggiore interesse nello sviluppo di sistemi automatizzati di raccolta di informazioni. La possibilità di estrapolazione ed analisi dati forniscono degli strumenti di lavoro applicabili in molti settori della società moderna. Si pensi ad esempio alla possibilità di impiego dei dati per analisi statistiche, per marketing, per pubblicità e per ricerche in genere.

La fruizione di dati relativi a singole persone, società, attività commerciali, contribuiscono alla alimentazione di sistemi di Big Data Analytics. Questi ultimi sono in grado di proporre strumenti di predizione ed analisi in chiave investigativa e di accelerare l'attività di Intelligence (nel particolare Virtual Intelligence).

Lo sviluppo di sistemi con un'operatività temporale continua, crea conseguentemente un'attenta risposta di contrasto. Ciò accade in quanto molto spesso l'impiego dell'attività di scraping avviene con fini malevoli e quindi considerata come "furto di dati". In particolare queste operazioni vengono esplicitamente proibite dai ToS [1] (termini di servizio) delle piattaforme web, soprattutto per ragioni di

[1] Terms of service

sicurezza delle informazioni e protezione dei dati pubblicati.

L'impiego di bot e botnet in grado di simulare e mascherare le operazioni automatizzate facendole apparire "umane", rappresenta la principale metodologia per lo scraping su larga scala. Precisamente l'avvio di un sistema distribuito avente più macchine con lo stesso obiettivo, permette di eludere agevolmente tutti gli eventuali controlli messi in atto dal fornitore dei servizi web per la protezione dei dati pubblicati.

La varietà di soluzioni di scraping attuabili presenta punti di forza per la gestione di operazioni massive. Diversificando metodologie di estrazione, i bot possono agire contemporaneamente e svolgere attività specifiche e diversificate, a seconda del target individuato.

Un esempio di raccolta dati automatica viene rappresentato dalla generazione e messa in campo di sistemi e strumenti di automazione dei browser. Questi sono in grado, anche visivamente, di simulare la normale attività di un utente sul web, premendo pulsanti, caricando siti internet e procedendo all'inserimento di input, oltre che ovviamente, al download delle informazioni di interesse.

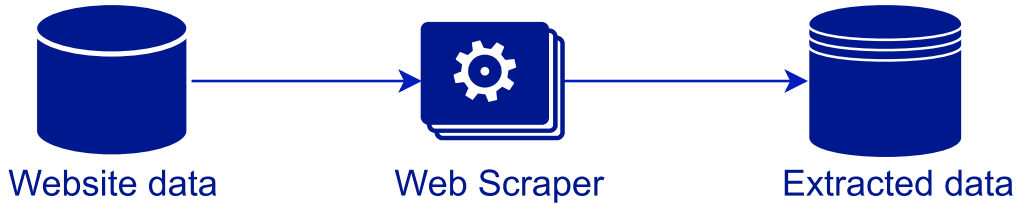


Figura 1: Schema di Web Scraping

2.1.1 L'estrazione dei dati dai Social Network

L'attività di scraping trova come importante campo d'applicazione i social network. Ad oggi, il vasto utilizzo di questi servizi porta alla registrazione di una moltitudine di dati diversi, caratterizzati dall'associazione degli stessi ad i singoli utenti. L'analisi della rete sociale di una persona, consente l'aggregazione dei dati e la sua conseguente profilazione. L'etereogeneità dei dati rappresenta il punto

iniziale per la catalogazione delle informazioni provenienti dalle piattaforme social. Il valore dei social network è direttamente proporzionale alla quantità di dati presenti.

L'attività denominata come "social engagement" ovvero il coinvolgimento degli utenti, fornisce un valido fattore di analisi sul comportamento che la singola persona attua di fronte ad un certo tipo di contenuto. Lo studio di queste informazioni garantisce una attenta profilazione dell'utente, molte volte inconsapevole della quantità di dati pubblici rilasciati dalla sua attività sui social.

Ogni piattaforma social a disposizione del pubblico adotta policy differenti riguardo alla estrazione dei dati. In particolare se da un lato vengono fornite Application Programming Interface appositamente sviluppate per la fruizione più semplice delle informazioni da parte degli sviluppatori, dall'altro si ha l'adozione di politiche restrittive, con API molto limitate ed un netto contrasto ai tentativi di estrapolazione dei dati. Negli ultimi anni, a seguito di clamorose notizie riguardanti l'utilizzo dei dati a discapito della privacy degli utenti, è diventato comune da parte delle piattaforme social l'introduzione di forti restrizioni sull'attività di scraping, attuando un vero e proprio contrasto a questo tipo di attività.

La presenza di un numero così grande di PII[2], ovvero qualsiasi rappresentazione di informazioni in grado di consentire l'identificazione di un individuo, ha generato maggiore attenzione sulla privatezza delle informazioni pubblicate dagli utenti, a tutela dei loro dati.[3]

L'identificazione di metriche idonee per lo sviluppo e l'analisi dei dati pubblicati si può classificare a seconda degli obiettivi di ricerca:

- **Volume dei dati**: un profilo con una grande influenza sul web può contenere fino a migliaia di contenuti di interesse operativo.

- **Testi**: i testi contenuti all'interno di post o all'interno di foto e video possono descrivere pensieri e sentimenti dell'utente.

[2]Personally identifiable information

- **Media**: le foto e i video pubblicati rappresentano un elemento fondamentale per l'analisi e la profilazione. L'eventuale presenza di luoghi o altre persone nei media garantisce una maggiore quantità di informazioni da immagazzinare ed analizzare.

- **Individui coinvolti**: l'azione del "tag" su una foto, video o su un post composto da solo testo, consente di ampliare la conoscenza su un utente, azionando il recupero delle informazioni anche su eventuali altri individui coinvolti.

- **Reazioni e commenti**: contribuiscono ad interpretare la diffusione di un messaggio o ideale propagato sul social network. Un alto numero è direttamente proporzionale all'interesse generato dal contenuto.

- **Geolocalizzazione**: la condivisione della posizione, correlata dalla specifica data ed ora, consente di identificare gli spostamenti o gli interessi di un utente, aumentando la precisione della sua eventuale profilazione.

- **Date e tempo**: l'estrazione di informazioni in un certo arco temporale d'interesse o la redazione di una totale cronostoria del profilo social favorisce l'individuazione di informazioni relative ad uno specifico periodo.

- **Rete sociale**: l'insieme degli amici o dei follower e dei seguiti, rappresenta la rete sociale dell'utente e gli eventuali interessi.

- **Gruppi di utenti**: la partecipazione a gruppi privati o pubblici presenta informazioni aggiuntive sull'individuo.

- **Pagine**: interesse verso pagine pubbliche di diversa tipologia (ad esempio notizie o brand).

- **Eventi**: partecipazione a manifestazioni, interesse nei confronti di eventi pubblici.

L'interpretazione di questi fattori, sia singolarmente che in gruppo individuano le informazioni di interesse per chi analizza i dati. Un numero maggiore di aggregazioni e correlazioni dei dati, genera una profilazione dell'utente più precisa ed utile.

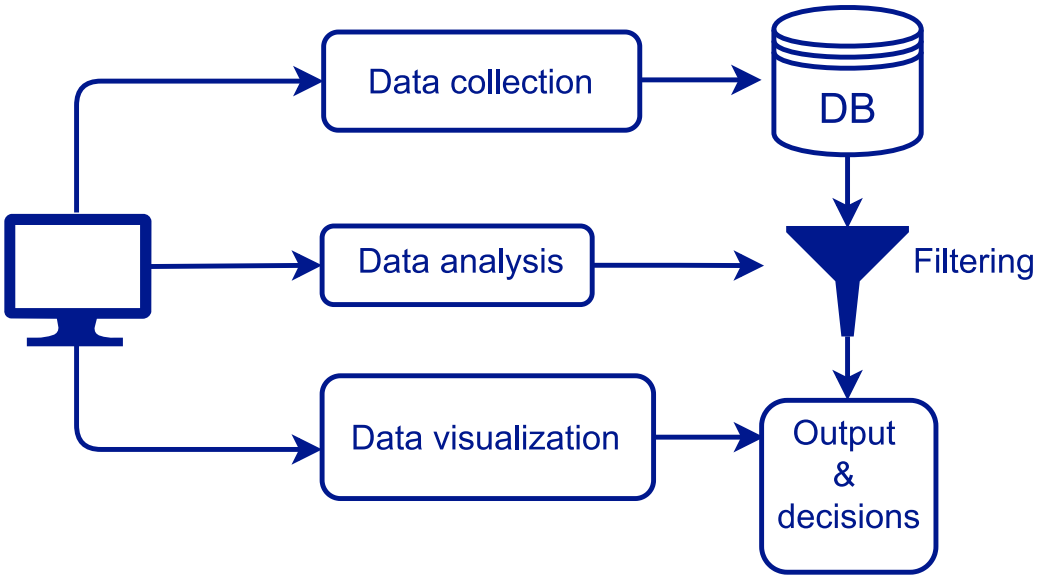


Figura 2: Schema di funzionamento di un sistema ideale di ricerca analisi ed estrazione dei dati da social network

2.1.2 Casi reali

Un esempio di notizia nota al pubblico per la forte influenza mediatica è il caso di Cambridge Analytica e Facebook. All'inizio dell'anno 2018 fu rivelato che la società Cambridge Analytica aveva effettuato una massiva raccolta di dati (circa 87 milioni di account del social Facebook) con conseguente profilazione degli utenti ed impiegato i risultati per attività di propaganda politica. La profilazione degli utenti avveniva attraverso l'utilizzo, da parte degli stessi, di un'app di "quiz sulla personalità", tramite la quale ciascun fruitore dava il consenso all'accesso ai dati del profilo.

La profilazione avveniva secondo un modello denominato OCEAN[3]: Openness (franchezza, apertura sociale), Conscientiousness (coscienziosità), Extraversion (estroversione), Agreeableness (gradevolezza, amicalità), Neuroticism (nevroticismo). Dal caso ne è derivata la chiusura dell'azienda per bancarotta, senza risvolti legali.[4]

Un altro caso di impiego attivo di web scraping è il software sviluppato dall'azienda americana Clearview AI. Il prodotto consente l'identificazione delle persona tramite l'intelligenza artificiale, sfruttando database composti anche da informazioni e foto estratte dai social network. Negli anni, i maggiori social hanno richiesto l'eliminazione dei dati estratti dalla società, anche se quest'ultima ha sempre dichiarato che l'impiego del progetto sia dedicato alle forze dell'ordine.

In Italia il Garante per la protezione dei dati personali, ha emesso in data 9 febbraio 2022 un'ordinanza di ingiunzione contro l'utilizzo di dati estratti da fonti pubbliche inerenti a cittadini italiani, sanzionando la società e vietandone la sua attività nello Stato Italiano. [4]

Altre implicazione legali sono state causate dal caso di Linkedin ed hiQ Labs. Nel particolare quest'ultima attuava soluzioni di web scraping sulla piattaforma Linkedin, raccogliendo informazioni sui profili registrati al servizio. La corte d'appello degli Stati Uniti ha dato ragione all'azienda hiQ Labs ritenendo che l'estrazione di dati pubblici non sia illegale. Di contro, Linkedin contrastava la decisione affermando la necessità di autorizzazioni per l'attività di scraping sulla loro piattaforma.

Le problematiche evidenziate dai casi descritti, hanno fatto sì che da parte delle piattaforme social, ci sia stato un conseguente forte incremento di soluzioni di sicurezza e contrasto al recupero ed estrazione dei dati non autorizzata.

[3]Anche denominata Teoria dei Big Five, è una tassonomia dei tratti di personalità

[4]`https://www.garanteprivacy.it/web/guest/home/docweb/-/docweb-display/docweb/9751362`

2.1.3 Metodologie di contrasto allo scraping

Attualmente, anche se con modalità differenti, ogni social adotta politiche di contrasto, ponendo come fulcro dell'attività la protezione dei dati personali dei propri utenti. La mitigazione del web scraping adottata dai social network individua delle tecniche con caratteristiche comuni:

- **Autenticazione ed accesso**: uno dei principali strumenti di deterrenza nei confronti dell'automazione dell'estrazione dei dati è l'obbligatorietà di iscrizione al sito web ed il conseguente login.

- **Ban**: attuazione di politiche di divieto d'accesso per dispositivi ed utenze identificate come "malevoli" e non rispettanti i termini di servizio. Il possibile ban si distingue in "soft-ban" e "permanent-ban", a seconda delle policy di contrasto del servizio. Per soft-ban si fa riferimento ad un divieto di durata limitata nel tempo, solitamente definibile in ore o giorni, mentre per permanent-ban si intende un divieto assoluto e permanente all'accesso da parte di un determinato utente.

- **Robots Exclusion Standard**: rappresenta un protocollo di comunicazione tra i siti internet ed i web crawler. Il metodo esclude i robot dall'accesso al sito attraverso l'espressione di politiche d'acesso (access policy) all'interno di un file denominato "robots.txt" ed accessibile sul server. Il file è caratterizzato da più record composti dalla descrizione dello "User Agent" e dalla regola "Disallow" (lett. "non consentire")[5].

  Si riporta un esempio estratto dal file robots di Facebook[6]:

```
User-agent: Googlebot
Allow: /*/videos/
Allow: /ajax/bootloader-endpoint/
Allow: /ajax/pagelet/generic.php/PagePostsSectionPagelet
```

[5] https://www.robotstxt.org/orig.html
[6] https://www.facebook.com/robots.txt

```
Allow: /careers/
Allow: /safetycheck/
Allow: /watch
```

- **Limitazione delle API**: le API messe a disposizione per gli sviluppatori presentano limitazioni sulla quantità dei dati estraibili, non consentendo uno scraping completo o massivo.
- **Limitazione delle richieste**: le richieste provenienti da uno stesso dispositivo ed in un ristretto periodo temporale vengono attenzionate ed impedite. L'analisi automatica del numero di richieste consente di gestire in modo meccanico i processi di ban.
- **Controllo indirizzi IP**: controllo sulla provenienza della richiesta. Viene effettuato un controllo sulle richieste provenienti dallo stesso indirizzo IP (o pool di indirizzi IP), iscrivendolo eventualmente in blacklist e bloccando la sua attività. Ciò può avvenire sia relativamente ai singoli IP, sia alla provenienza da regioni geografiche simili o vicine.
- **Browser fingerprinting**: analisi remota delle informazioni di un dispositivo. Le informazioni del client ricavabili con questa tecnica indicano il browser utilizzato (con versione e personalizzazioni), sistema operativo e applicazioni, hardware e network.[5]
- **Identificatori pseudoanonimi**: l'attività di scraping si avvale dell'estrazione diretta dei dati a partire da un link, molte volte statico. Per evitare ciò viene prevista la presenza di ID pseudoanonimi[7] random generati a partire dall'ID fisso ed il timestamp[8]. Questo previene l'attività dei tool che sfruttano la staticità dei link per avviare le operazioni di GET.

[7]Si identificano come informazioni non completamente anonime

[8]Anche denominata marca temporale, è una sequenza di caratteri che rappresentano una data e un orario ed in grado di accertare l'avvenimento di un evento

- **Identificazione "umana"**: alcuni social network per evitare che venga effettuato un login automatico da parte di bot, obbligano all'identificazione di una persona reale, attraverso una foto da scattare sul momento.
- **Aggiornamenti**: continuo sviluppo di metodi di contrasto allo scraping. Creazione di team di sviluppo appositamente dedicati a questa funzione.

L'elenco delle attività sopra descritte rappresenta una sola parte dei provvedimenti che ad oggi vengono adottati per contrastare l'automazione nella raccolta dei dati. È importante rendere noto che per la maggior parte dei servizi web, la risposta allo scraping non implica solo una delle soluzioni descritte bensì l'insieme di almeno due o più "ostacoli" di tipo tecnico.

2.2 L'utilizzo dei dati

2.2.1 Open Source Intelligence

L'OSINT [6] (Open source intelligence) rappresenta il ramo dell'intelligence che si occupa della ricerca e catalogazione delle informazioni da fonti aperte. Per fonti aperte si intendono i media, i dati pubblici ed i dati accademici. Questo tipo di attività sviluppa come vantaggio l'accessibilità delle informazioni, le quali essendo pubbliche, risultano facilmente reperibili; mentre la quantità di dati da gestire rappresenta il principale problema di questa tecnica. L'applicazione dell'attività di intelligence su piattaforme social network prende il nome di "Social media intelligence". [7].

L'etereogeneità dei dati pubblicati da un utente sulle piattaforme social, consente all'analista d'intelligence lo sviluppo di schemi di analisi comportamentali e sociali. La presenza di dati temporanei ("a scadenza"), come ad esempio le storie, fornisce informazioni in tempo reale sull'attività di un soggetto sottoposto a controllo. Il tracciamento e la profilazione di un individuo si completa grazie ad informazioni che esso stessa pubblica come la geolocalizzazione, la sua rete sociale, i post e i media, le reazioni ed i commenti, oltre che agli interessi personali espressi tramite i "mi piace" e la sua attività nel complesso. È importante fare presente

che la principale fonte dell'OSINT non è rappresentata dal Worldwide Web ma dal Deep Web, ovvero la porzione di web non indicizzata dai motori di ricerca.

2.2.2 Scenari d'impiego

I dati raw, "grezzi", estratti dalle fonti pubbliche, una volta sottoposti ad ingestione in sistemi di Big Data Analytics e processati, possono essere analizzati tramite tecniche di Sentiment Analysis[9]. I dati prodotti, descritti al Paragrafo 2.1.1, possono essere impiegati in molteplici attività come:

- **Contrasto a radicalizzazione e terrorismo online**: le policy dei social network vietano la pubblicazione di contenuti relativi ad organizzazione terroristiche o a relative radicalizzazioni e di contenuti violenti, attuando, anche tramite lo strumento della segnalazione, l'eliminazione di questi dati. Ciò non avviene sempre in maniera immediata o corretta e per questo motivo, garantire un'estrazione dati continua e performante consentirebbe di ricavare informazioni che in futuro saranno sottoposte a controllo ed eliminazione. L'impiego da parte di gruppi sovversivi di piattaforme web di interazione sociale, facilita anche il processo di reclutamento, promuovendo ideali ed azioni in contrasto con la società moderna. Ciò consente la decentralizzazione dell'organizzazione, generando un'engagement[10] transnazionale, ampliandosi facilmente e di diffondendo la dottrina e le strategie sovvertitrici. [8] Garantire uno strumento operativo in grado di "intercettare", prevedere e consultare informazioni relative a probabili associazioni di tipo terroristico può rappresentare un valore aggiunto per le agenzie internazionali e i reparti delle Forze dell'Ordine che si occupano di queste tematiche.

- **Investigazione**: come introdotto al Paragrafo 2.2.1, i dati pubblici rappresentano la chiave di successo in un'attività di Open Source Intelligence. I dati eterogenei provenienti da uno specifico utente permettono la correlazione con casi di studio in fasi investigative. La quantità di informazioni presenti

[9] Elaborazione del linguaggio naturale per l'estrazione di opinioni e sentimenti.
[10] Coinvolgimento, impegno

consente una maggiore attività d'indagine in grado di fornire strumenti a supporto delle operazioni.

- **Ricerca**: l'analisi dei dati consente di affinare metodologie di ricerca ed interpretazione delle relazioni sociali e delle informazioni condivise pubblicamente. La ripetizione di azioni e la maggiore frequenza di condivisioni di determinate comunicazioni garantiscono il maggiore coinvolgimento del pubblico, inteso come rete sociale di un utente.

- **Strumenti di supporto**: la creazione di strumenti a supporto delle tematiche sopra riportate possono generare una soluzione per le attività delle Agenzie di Sicurezza a livello internazionale, che si occupano della catalogazione ed analisi dei dati per diversi scopi.

2.3 Privacy e dati pubblici

2.3.1 Aspetti legali

La legalità dell'attività di web scraping è un aspetto controverso e citato in diversi casi reali, come proposto al Paragrafo 2.1.2.

Il caso rientra in una "zona grigia" in ambito giuridico, in quanto la fattispecie non è espressamente definita da nessuna norma.[9] L'attività risulta essere trasversale ad una serie di teorie legali, leggi internazionali e di comune applicazione come:

- **Termini di utilizzo**: descrivono le condizioni di utilizzo e le relative attività vietate dalla piattaforma che ospita i dati, attuando soluzioni tecniche di contrasto, come riportato al Paragrafo 2.1.3. L'accettazione dei termini di utilizzo di un servizio web è comunemente obbligatoria per l'accesso allo stesso.

- **Copyright**: estrarre gli elementi coperti da copyright e ripubblicarli viola le fattispecie identificate dall'istituto giuridico della tutela di un'opera. A seconda del paese, l'attività ricade in legislazioni differenti.

- **Danneggiamento**: l'impiego di tool con operatività continua ed in grado di effettuare azioni massive può portare al danneggiamento del server su cui si opera. Anche in questo caso, a seconda del paese, la fattispecie viene identificata in regolamenti differenti. Ad esempio in Italia, l'azione può ricadere nell'Art.635 bis del Codice Penale, ovvero "Danneggiamento di informazioni, dati e programmi informatici".

- **Scopo dell'attività**: in generale ogni impiego non legale o fraudolento dei dati ricavati è punito da diverse leggi a livello internazionale. Ad esempio negli Stati Uniti, secondo il Computer Fraud and Abuse Act[11], l'attività di recupero di dati pubblici (web scraping in generale) è ritenuta legale, ma è punito l'impiego di dati per scopi non autorizzati.

In Europa, la presenza del GDPR (General Data Protection Regulation)[10] e di vari casi di applicazione recenti, definiscono una direzione univoca nei confronti del trattamento dei dati non autorizzato, come accade nel web scraping. Si riportano le definizioni di interesse espresse nel testo del Regolamento UE:

Art.4 comma 1

"«dato personale»: qualsiasi informazione riguardante una persona fisica identificata o identificabile («interessato»); si considera identificabile la persona fisica che può essere identificata, direttamente o indirettamente, con particolare riferimento a un identificativo come il nome, un numero di identificazione, dati relativi all'ubicazione, un identificativo online o a uno o più elementi caratteristici della sua identità fisica, fisiologica, genetica, psichica, economica, culturale o sociale."

[11] CFAA, Computer Fraud and Abuse Act, 1986, legge federale degli Stati Uniti d'America

Art.4 comma 2

"«trattamento»: qualsiasi operazione o insieme di operazioni, compiute con o senza l'ausilio di processi automatizzati e applicate a dati personali o insiemi di dati personali, come la raccolta, la registrazione, l'organizzazione, la strutturazione, la conservazione, l'adattamento o la modifica, l'estrazione, la consultazione, l'uso, la comunicazione mediante trasmissione, diffusione o qualsiasi altra forma di messa a disposizione, il raffronto o l'interconnessione, la limitazione, la cancellazione o la distruzione."

Art.4 comma 4

"«profilazione»: qualsiasi forma di trattamento automatizzato di dati personali consistente nell'utilizzo di tali dati personali per valutare determinati aspetti personali relativi a una persona fisica, in particolare per analizzare o prevedere aspetti riguardanti il rendimento professionale, la situazione economica, la salute, le preferenze personali, gli interessi, l'affidabilità, il comportamento, l'ubicazione o gli spostamenti di detta persona fisica."

Gli articoli riportati, individuano gli aspetti legali in cui si incorre utilizzando e sviluppando strumenti di scraping. A partire dal trattamento in genere, fino alla profilazione, l'attività potrebbe incorrere in casi legali qualora non sia esplicitamente consentita dal proprietario dei dati estratti o da chi ne ha la responsabilità.

Particolare attenzione viene posta da parte del GDPR, ove all'articolo 6, viene prevista la liceità del trattamento in determinati e specifici casi.
È importante fare presente che in Italia esiste una fattispecie di reato denominata "Comunicazione e diffusione illecita di dati personali oggetto di trattamento su larga scala" (art. 167-bis Codice della Privacy, D.lgs. 30 giugno 2003, n. 196) facente riferimento all'utilizzo dei dati "su larga scala", come può esserlo tramite l'attività di web scraping massiva.
Si riporta il testo dell'articolo.

Art.167-bis Codice della Privacy

Comma 1. *Salvo che il fatto costituisca più grave reato, chiunque comunica o diffonde al fine di trarre profitto per se' o altri ovvero al fine di arrecare danno, un archivio automatizzato o una parte sostanziale di esso contenente dati personali oggetto di trattamento su larga scala, in violazione degli articoli 2 ter, 2 sexies e 2 octies, è punito con la reclusione da uno a sei anni.*

Comma 2. *Salvo che il fatto costituisca più grave reato, chiunque, al fine trarne profitto per sé o altri ovvero di arrecare danno, comunica o diffonde, senza consenso, un archivio automatizzato o una parte sostanziale di esso contenente dati personali oggetto di trattamento su larga scala, è punito con la reclusione da uno a sei anni, quando il consenso dell'interessato è richiesto per le operazioni di comunicazione e di diffusione.*

L'applicazione della disciplina di protezione dei dati viene esclusa nel caso di trattamento dati per motivi di giustizia, di sicurezza e per finalità di prevenzione e repressione dei reati.

Capitolo 3

Analisi concettuale

3.1 L'estrazione dei dati

3.1.1 Requisiti

Il fine del progetto è quello di fornire un servizio portabile che effettui automaticamente lo scraping di dati, in modo massivo, da fonti Social Network. Il software deve garantire un'operatività continua anche a fronte di eventuali problemi relativi al contrasto delle operazioni o inconvenienti tecnici. Più informazioni estratte dalle fonti implicano maggiori generazioni di modelli di previsioni ed analisi. Obiettivo operativo del software da produrre è la quantità effettiva di dati e la resistenza in campo d'azione. La portabilità è garantita attraverso lo sviluppo di API e la gestione del software in container. I dettagli tecnici in riferimento alle tecnologie da impiegare sono descritti al Paragrafo 4.5.

Le caratteristiche di un software di scraping ideale riuniscono in questi requisiti:

- **Portabilità**: impiego del software su piattaforme diverse da quella di origine.
- **Interoperabilità**: capacità del software di scambiare informazioni provenienti da più fonti (es. diversi social network) ed essere in grado di utilizzarle.
- **Resilience to failure**: capacità dell'applicazione di scraping di mantenere l'operatività nonostante eventuali problemi tecnici o azioni di contrasto. È importante sviluppare un software resiliente e continuamente aggiornato,

osservata l'attenzione da parte dei fornitori dei dati, alla mitigazione dello scraping.

- **Gestione dei dati**: garantire l'uniformità dei formati dei dati, qualora essi provengano da diverse fonti per agevolare la data ingestion.

- **Data ingestion**: processo di trasporto dei dati, necessario per fornire le informazioni ai sistemi di analytics. Gestire un trasporto rapido, veloce e sicuro, consente di apportare migliorie al funzionamento totale del software.

3.1.2 Infrastruttura

Per ogni social network oggetto di studio viene sviluppato un connettore che rappresenta idealmente un canale di estrazione ed ingestione dei dati. Ogni connettore ha caratteristiche differenti a seconda del social individuato come obiettivo. La struttura software della piattaforma target è varia e per questo il connettore ed il suo "motore" di scraping presentano proprietà diverse. Il punto in comune tra i vari scraper è la gestione degli output. Osservato che il canale di immissione dati è unico, di conseguenza ogni connettore deve aderire alle policy del sistema di Big Data. Il singolo data connector ha quindi il compito di estrarre e presentare i dati per il successivo step di analisi.[11] Questa attività rappresenta il fulcro dell'intero progetto e si scontra con le principali problematiche già descritte come il contrasto tecnico allo scraping e le fattispecie legali. Il funzionamento tecnico dei connettori individua azioni di estrazione specifica dal target. La ricerca dei dati può essere effettuata attraverso gli hashtag, i nomi utente, i gruppi e le pagine. Tutto l'output prodotto verrà successivamente immesso nel sistema di analisi. L'estrazione di un singolo profilo di un utente individua tutta la sua storia sul social, a partire dalla prima informazione pubblicata, fino al giorno in cui avviene l'operazione.

3.1.3 Representational state transfer

Il paradigma REST (representational state transfer) definisce lo stile architetturale per sistemi distribuiti, è basato su HTTP ed si basa su quattro principi:

- identificazione delle risorse tramite Uniform Resource Identifier (URI);

- presenza un'interfaccia uniforme per tutte le risorse;
- rappresentazione di risorse e metadati;
- presenza di collegamenti ipertestuali.

I punti di forza dell'architettura REST si basano sull'assenza di sessione (stateless), sulla scalabilità e sul largo impiego nelle applicazioni web.

L'architettura REST permette di utilizzare i metodi HTTP[1] per la gestione delle risorse.[12] In particolare, nel processo di acquisizione dati è stato utilizzato il metodo HTTP GET per restituire elenchi e rappresentazioni delle informazioni. Questa azione rappresenta il punto cardine dell'attività di scraping.

3.1.4 Application Programming Interface

Un'interfaccia di programmazione di un'applicazione consiste in un insieme di definizioni, protocolli e procedure messo in atto per favorire l'integrazione e la comunicazione tra diverse applicazioni. In particolare, un'API è detta RESTful quando si attiene ai criteri del paradigma REST citati nel paragrafo 3.1.3.

Le API REST basano totalmente il loro funzionamento su HTTP, includendone le modalità di richiesta, risposta ed intestazione dei messaggi.[13]

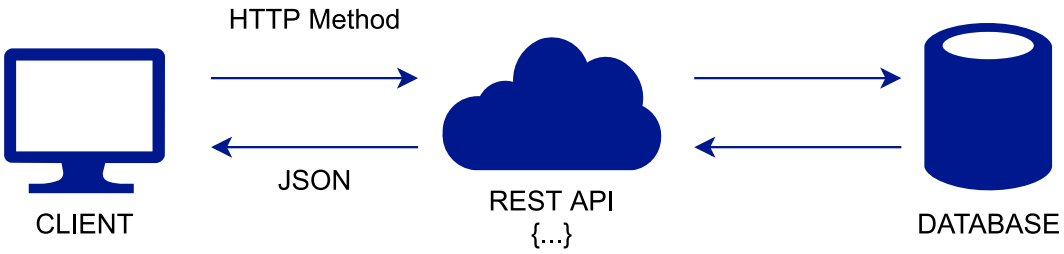


Figura 3: Schema di funzionamento API

[1]HyperText Transfer Protocol

3.1.5 Software container

Un software container, lett. "contenitore di software", fornisce un ambiente di esecuzione per le applicazioni che condividono un sistema operativo host, codice e librerie con altri container. [14] Il vantaggio principale dell'utilizzo dei container è la portabilità dello stesso, in quanto può essere installato ed eseguito su qualsiasi ambiente IT. Il servizio da implementare è Docker[2]. Docker, una piattaforma che garantisce il funzionamento delle applicazioni in qualsiasi ambiente e composta da quattro parti:

- **Docker Engine e Docker Client-Server**: esegue processi secondo l'architettura Client-Server, attraverso un server che opera senza termine (daemon), API Rest per la comunicazione tra i componenti e un client che effettua le richieste;
- **Docker Image**: l'immagine è composta da più strati, a partire dalla base caratterizzata da una distribuzione Linux leggera[3]. Con la generazione di un Dockerfile e lanciando il comando "docker build" viene generata l'immagine;
- **Docker Container**: il container viene generato a partire dall'immagine ed include tutti gli strumenti necessari per avviare ed eseguire l'applicazione.

Docker consente di gestire le risorse e standardizzare le applicazioni sviluppate, offrendo soluzioni di distribuzione, prevedendo un eventuale sviluppo di progetto verso l'implementazione di sistemi distribuiti.[15] Si riporta lo schema[4] della struttura di un container Docker.

[2] `https://www.docker.com/`

[3] Ubuntu, Fedora, CentOS

[4] Schema tratto da `https://docker.com`

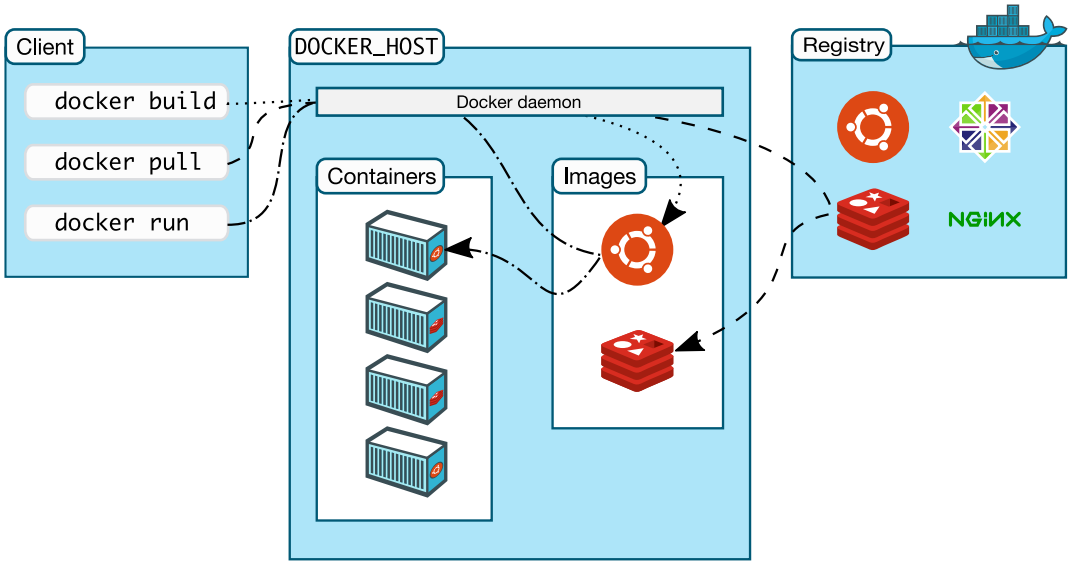


Figura 4: Struttura di un container Docker

3.1.6 Cookies

I cookies rappresentano dei piccoli blocchi di dati, generati lato server e memorizzati sul client, in grado di espletare varie funzioni. Nella loro configurazione è possibile definire diversi attributi a seconda dello scopo di utilizzo degli stessi.

La diversificazione nel loro impiego, identifica diverse tipologie di cookies:

- **Cookie di sessione**: vengono generati nel momento in cui l'utente effettua operazioni di navigazione web, scadono una volta che viene chiuso il browser.
- **Cookie persistenti**: basano il loro funzionamento su uno specifico arco temporale. In particolare vengono memorizzati lato client e trasmessi nel caso in cui si ha la necessità del loro impiego in un determinato sito web. Una volta terminato il periodo di validità scadono e devono essere generati nuovamente. Questa tipologia di cookie viene adottata dai siti web in cui è presente un form di autenticazione, mantenendo in questo modo il login.
- **Cookie di terze parti**: tracciano l'attività dell'utente attraverso la navigazione nei siti web. Vengono impiegati in ambito pubblicitario e generano il collegamento a server differenti da quello in cui si sta fruendo un determinato servizio, per recuperare informazioni memorizzate durante tutta la sessione di navigazione.

L'utilizzo dei cookie, impone una maggiore gestione della sicurezza, soprattutto nella memorizzazione dati relativi a sessioni di login.[16] I cookie di tipologia persistente, vengono impiegati durante l'accesso ai social network; il salvataggio di questi dati, permette di effettuare login ai servizi, senza l'utilizzo di credenziali.

3.1.7 L'ingestione dei dati

Per Data Ingestion si intende il processo di raccolta dei dati raw (grezzi) dalla loro sorgente e del trasporto e la centralizzazione verso il sistema target.[17]

Viene definito anche ETL (Extract/Transform/Load) ovvero estrazione, trasformazione e caricamento. Le tre fasi funzionali assolvono i seguenti compiti:

- **Estrazione**: la prima fase attua le tecniche definite al Capitolo 2. Le soluzioni adottate per l'elusione dei controlli e la mitigazione del contrasto saranno dettagliate al Paragrafo 3.2.

- **Trasformazione**: la seconda fase si occupa del controllo e del data cleaning, ovvero il processo in grado di garantire l'affidabilità dei dati, anche trasformandoli in formati idoeni al loro utilizzo. Nel caso del progetto di tesi è stato scelto il formato JSON, dettagliato al Paragrafo 4.5.1.

- **Caricamento**: la terza e ultima ultima fase rappresenta il caricamento dei dati nel sistema di Big Data Analytics. La gestione di questo step è garantita attraverso l'omogeneità negli output e nelle tecnologie richieste dal target per ogni connettore.

3.1.8 Big Data Analytics

Con il termine Big Data si definisce un sistema avente tre attributi: grande volume di dati, varietà e velocità. Il volume rappresenta l'attributo principale, seguito dall'eterogeneità e dalla frequenza di generazione e rilascio dei dati.

L'applicazione del termine "Big Data Analytics" comprende tecniche avanzate di analisi dei dati su set di Big Data.[18] L'analisi dei dati si distingue a seconda degli strumenti e dei modelli di gestione in tre tipologie:

- **Analisi descrittiva**: descrizione storica e confronto dei dati presenti. Tecnica da impiegare per la catalogazione delle informazioni attuali e passate.
- **Analisi predittiva**: applicazione di modelli predittivi sui dati, in grado di estrarre e prevedere casi futuri.
- **Analisi prescrittiva**: propone soluzione automatizzate ed implicazioni di azioni in corso sui dati.

L'insieme dei tre modelli di analisi sono applicabili per lo scopo del progetto ed il conseguente studio delle informazioni pubbliche estratte dai social.

3.2 Soluzioni al contrasto dello scraping

3.2.1 Generalità

Le metodologie attuate dalle piattaforme social per contrastare l'attività di scraping sono state elencate e descritte al Paragrafo 2.1.3. Si presentano di seguito soluzioni ideali per garantire il funzionamento di un'applicazione resiliente ed in grado di offrire l'estrazione della maggiore quantità di dati.

3.2.2 Gestione degli account

I social network, per la normale attività sulle loro piattaforme, obbligano ad effettuare la registrazione ed il conseguente login. Le possibilità di accesso ai servizi senza alcuna tipologia di autenticazione sono molto limitate e non giovano in alcun modo ad una completa estrazione dati. Da ciò ne consegue la necessità di autenticarsi ai servizi e la creazione di un pool di account. L'idoneità di un account risulta essere oggetto di attenzione dai servizi web dei social. Le soluzioni di sicurezza "anti-bot"[5] sono in grado di individuare la generazione di utenti falsi, definiti "fake accounts" promuovendo ed attuando il loro ban.

[5]Con il termine "bot" si fa riferimento ad un servizio che compie azioni in modo automatico

3.2.3 Rotazione dei Cookies

Una volta avuta la disponibilità di un numero cospicuo di account validi, si procede con l'azione di login e la successiva estrazione dei cookie, per ogni utenza. L'estrazione dei cookie può avvenire attraverso l'accesso diretto dal tool di scraping, oppure tramite lo sviluppo di azioni in funzione di applicazioni di browser automation. La seconda opzione risulta essere maggiormente indicata, soprattutto in visione della scadenza di validità temporale dei singoli cookie, che rende obbligatoria la ripetizione del login.

Un recupero della sessione automatico e il conseguente aggiornamento dei cookie consente la validità continua degli stessi. Una soluzione implementativa idonea all'elusione del controllo sullo scraping è l'introduzione, dopo aver memorizzato tutti i cookie validi degli account, di un sistema di rotazione random sulla selezione ed impiego del cookie appartenente ad un determinato utente. Questa operazione consente di diversificare gli accessi automatici al social, limitando fortemente la probabilità di ban di un account, osservato che lo stesso verrà impiegato per una singola operazione alla volta e successivamente sostituito da un altro nell'elenco, diminuendo così l'esposizione del singolo.

Ne consegue che un maggior numero di account validi disponibili per l'accesso al social network, consente una massimizzazione dell'elusione al controllo sul singolo utente e la sua attività. Avendo più cookie in grado di essere selezionati dalla funzione random, viene aumentata l'operatività continua, in senso temporale, dell'estrazione dei dati.

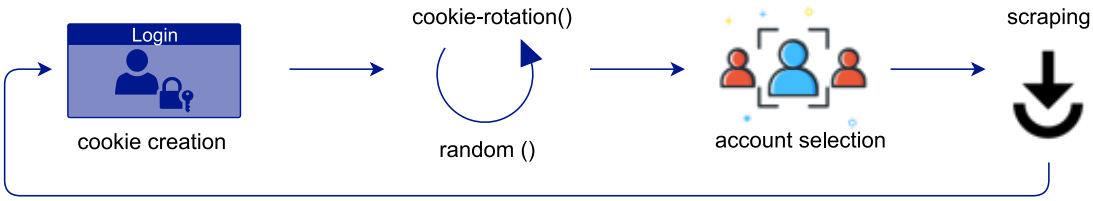


Figura 5: Cookie Rotation

3.2.4 Latenza

Il contrasto alle soluzioni anti-scraping richiede anche un tempo operativo più dilatato. Si rende necessaria l'aggiunta di latenza tra le operazioni in termini di secondi o minuti, garantendo un profilo di attività non sospettoso. L'aggiunta di tempo d'attesa tra le operazioni consente la simulazione dell'attività umana sulla piattaforma web, non dimostrando l'automaticità e la velocità tipica dei bot. Un'idea implementativa è l'introduzione di attesa random nell'ordine dei minuti. Alcuni tool di scraping prevedono automaticamente una funzione di salvaguardia dell'account, bloccando le richieste ed azionando un'attesa a seconda dell'onerosità dell'attività da svolgere. Considerato che l'obiettivo del progetto è la garanzia operativa senza l'effettiva necessità di velocità nella produzione dei dati, il ritardo tra le azioni non vincola l'impiego del software da produrre.

3.2.5 User Agent

Lo user agent identifica un campo del protocollo HTTP in cui vengono dichiarate delle informazioni relative al dispositivo che si connette al server ed il browser utilizzato. Un esempio di user agent valido per la connessione ad Instagram e relativo ad un dispositivo mobile iOS è il seguente:

Mozilla/5.0 (iPhone; CPU iPhone OS 15_5 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Mobile/15E148 Instagram
244.0.0.12.112 (iPhone12,1; iOS 15_5; en_US; en-US; scale=2.00; 828x1792;
383361019)

Nell'atto dello scraping è importante definire user agent idonei ed approvati dal servizio target in quanto lo stesso può effettuare specifici controlli e non accettare le richieste provenienti da user agent non presenti in "white-list"[6].

[6]Si fa riferimento all'elenco degli user agent accettati dal servizio.

3.2.6 Indirizzi IP

I sistemi anti-scraping pongono particolare attenzione agli indirizzi IP dai quali provengono le richieste. Come introdotto al Paragrafo 2.1.3, viene attuato un controllo sulla affidabilità dell'indirizzo da cui si è connesso un determinato utente. Nel caso in cui venissero rilevate delle azioni "insolite" o massive, si procede con il ban temporaneo o permanente, a seconda delle policy previste dal servizio.

Un'idea implementativa è l'utilizzo di un pool di indirizzi IP da proxy condivisi. L'introduzione di una "ip-rotation" randomica, seguendo quanto progettato per la cookie-rotation (Paragrafo 3.2.3), può diminuire il numero di ban e garantire un maggiore impiego delle utenze a disposizione.

Si riporta uno schema per il funzionamento di "ip-rotation".

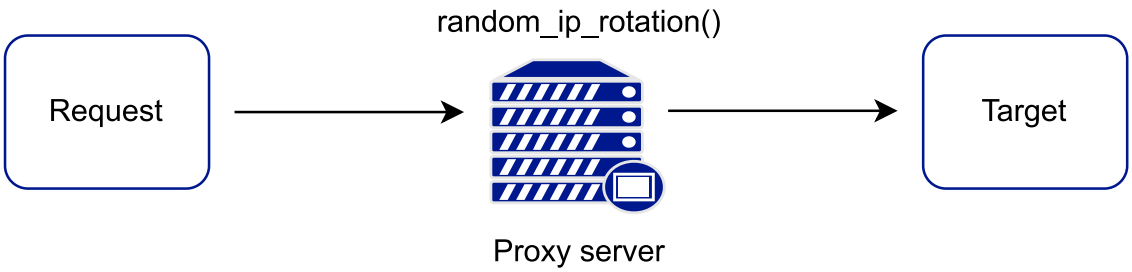


Figura 6: IP Rotation

3.2.7 Sistemi distribuiti

Un sistema distribuito è un insieme di processi indipendenti ed interconnessi che cooperano per la condivisione di risorse. [19]

Le caratteristiche dei sistemi distribuiti sono:

- **Openess**: possibilità di estendere, in termini di risorse, un sistema.
- **Concurrency**: presenza di più processi coesistenti su un'unica risorsa.
- **Scalability**: capacità del sistema di aumentare ed adattarsi all'aumento dimensionale del lavoro da compiere senza alterare la funzionalità.
- **Fault tolerance**: tollerenza del sistema al guasto.

L'implementazione della distribuzione su un'applicazione di scraping consente di ottenere molteplici vantaggi oltre a quelli propri dei sistemi distribuiti. Può essere identificata come una soluzione generalizzata al contrasto dello scraping, in quanto consente di aggiungere funzionalità utili all'elusione dei controlli. La funzionalità di un sistema distribuito per lo scraping necessita della presenza di un controller in grado di inviare comandi ai singoli nodi e ricevere risposte in termini di dati estratti dai singoli scraper-bot. La comunicazione tra controller e singoli bot può avvenire tramite Remote Procedure Call (RPC) [7] Si presentano di seguito gli schemi di funzionamento e comunicazione delle varie parti in azione durante lo svolgimento di un'attività di scraping distribuita.

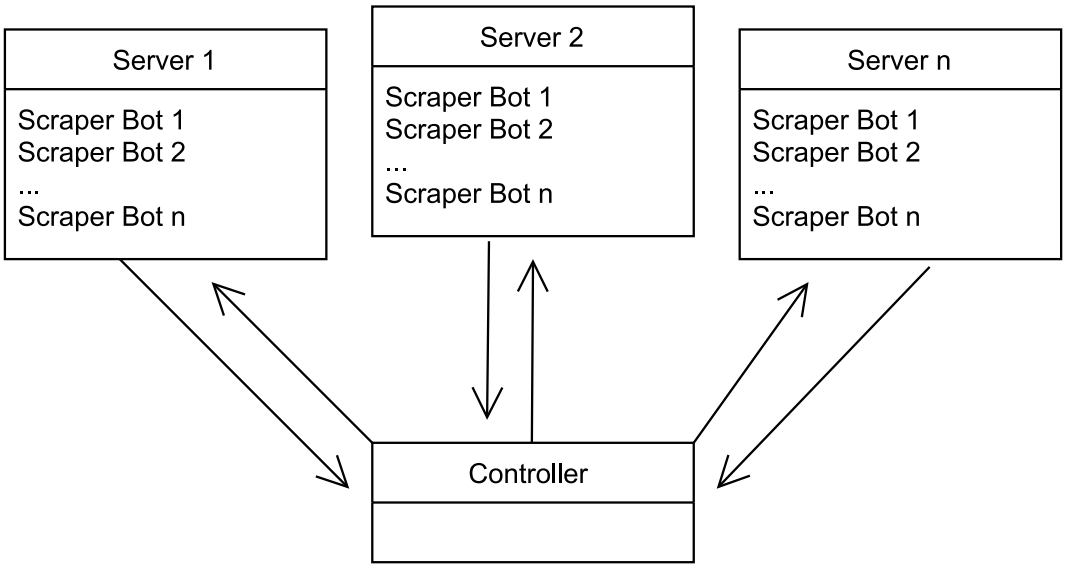


Figura 7: Schema di un'applicazione di scraping distribuita

[7]Indica una procedura avviata su dispositivo diverso da quello sul quale viene eseguito un programma.

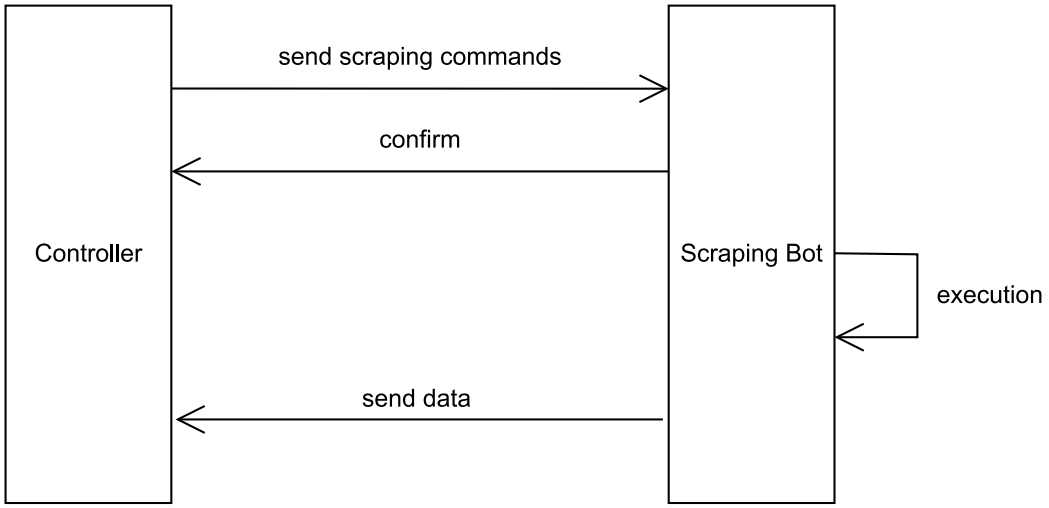


Figura 8: Schema di comunicazione controller-bot

3.3 Output

3.3.1 Gestione dei dati estratti

I dati in output devono essere uniformi e pronti all'ingestione in un sistema di analisi. Per garantire l'uniformità si devono individuare formati e tipi di dato uguali, a fronte dell'estrema eterogeneità dei dati raccolti. Un esempio è l'introduzione dell'utilizzo di UUID (Universally Unique Identifier) per la denominazione dei file e la compressione delle cartelle in formato zip per la riduzione dello spazio di memoria occupato. UUID rappresenta un identificativo univoco universale composto da 128 bite rappresentato da caratteri esadecimali divisi in gruppi. Si fa presente che i dati effettivamente estratti restano in formati direttamente fruibili quali JSON, JPEG e MP4. L'impiego di UUID e zip ha come obiettivo la sola modifica della denominazione e della dimensione dei file per la loro successiva ingestione ed analisi come Big Data.

Diversi tool offrono la possibilità di individuare file già presenti in locale, frutto di estrazioni precedenti, evitando di effettuare un nuovo download. Questa funziona viene adottata anche per individuare eventuali cambiamenti "storici" nell'account target: nel caso in cui un elemento già estratto in passato non venga più selezionato e scaricato, si deduce automaticamente la presenza di un'azione di cancellazione dello stesso dalla piattaforma social.

L'analisi comportamentale del software rispetto a cambiamenti storici nell'output garantisce un elevato numero di informazioni aggiuntive che vanno ad ampliare i metadati già frutto di estrazione. Le stesse, combinate con il resto, garantiscono una continuità informativa nell'atto dell'ingestione dei dati nel sistema di Big Data Analytics.

Si riporta di seguito uno schema di gestione dei dati di output provenienti dal software di scraping.

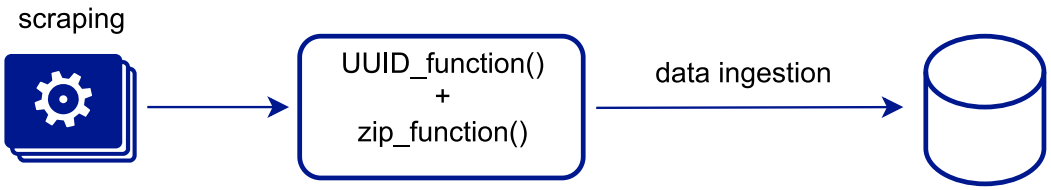


Figura 9: Schema di gestione dei dati per l'ingestione

Capitolo 4

Il caso di studio

4.1 I social di Meta

Per il progetto sono stati individuati come obiettivi i due principali social network della società Meta Platforms Inc.[1], ovvero Facebook ed Instagram. Le due piattaforme sono state selezionate principalmente per il vasto pubblico attivo e conseguentemente per la quantità di informazioni online.

4.1.1 Facebook

Facebook, nato nel 2004, conta ad oggi secondo i dati ufficiali rilasciati dalla società, 2.93 miliardi di utenti attivi [2]. Il social permette di condividere informazioni testuali e multimediali sia pubblicamente che privatamente, attraverso singole pagine personali, pagine pubbliche e gruppi. La funzionalità denominata come "amicizia" consente la creazione di una rete sociale per ogni singolo utente. Le foto, i video, i post, le storie, le reazioni, i commenti, la condivisione di contenuti in diretta, la presenza di un market ed i messaggi istantanei sono i principali punti di forza della piattaforma. L'imponente diffusione di questo social ha implicato l'impiego dello stesso anche come strumento informativo ufficiale da parte

[1] https://about.meta.com

[2] https://investor.fb.com/investor-news/press-release-details/2022/Meta-Reports-Third-Quarter-2022-Results/default.aspx

di organi statali, di stampa e di informazione pubblica. L'aggiornamento costante del servizio aggiunge continuamente nuove funzionalità che vanno ad ampliare la quantità di informazioni pubblicate.

4.1.2 Instagram

Instagram, pubblicata nel 2010, raggiunge ad oggi quasi lo stesso numero di utenti attivi di Facebook. I punti di forza di questo social, nato inizialmente per condividere solamente foto, riuniscono varie possibilità di pubblicazione di informazioni online. Oltre alle semplici foto, possono essere pubblicati altre tipologie di contenuti multimediali come video e musica. La piattaforma consente la pubblicazione dei media sotto forma di post, permettendo agli utenti di cliccare "mi piace" e scrivere commenti. Inoltre il singolo utilizzatore può generare la sua rete sociale tramite il "follow" lett. "seguire" ed attraversp l'utilizzo il sistema di messaggistica istantanea e la condivisione di storie e post. I profili degli utenti possono essere impostati come pubblici o privati, questi ultimi necessitano l'autorizzazione del proprietario per essere consultati. Anche nel caso di Instagram, l'aggiornamento è costante in quanto vengono rilasciate costantemente nuove funzionalità a favore degli utenti.

4.1.3 Termini e condizioni

Le piattaforme sopra introdotte individuano termini e condizioni d'utilizzo simili. Nel particolare, in riferimento alla finalità del progetto sviluppato, viene esplicitamente vietato qualsiasi comportamento riconducibile ad azioni di web scraping o più in generale di collezione dei dati in maniera automatica.[20] Come riportato dal file "robots.txt"[3] di Facebook:

> *"Notice: Collection of data on Facebook through automated means is prohibited unless you have express written permission from Facebook and may only be conducted for the limited purpose contained in said permission. See: http://www.facebook.com/apps/site_scraping_tos_terms.php"*

[3] `https://instagram.com/robots.txt`

Meta contrasta la External Data Misuse (EDM)[4] promuovendo nuove tecniche contro lo scraping dei dati. Un deterrente attuato dalla società è anche la promozione di azioni legali contro attività non consentite dai termini di servizio. Pubblicando notizie ed aggiornamenti su come viene contrastata la collezione automatica di dati si punta a dissuadere un eventuale soggetto interessato all'attività.

4.2 Strumenti Open Source per lo scraping

Il funzionamento dei singoli connettori di scraping si basa su due tool open source individuati come migliori per lo scopo designato. In particolare, per i due social network oggetto di studio, Facebook e Instagram, si è fatto uso rispettivamente di "facebook-scraper"[5] e "instaloader"[6].

L'individuazione e lo studio di questi strumenti, si è basato su un attento confronto delle loro caratteristiche tecniche e di funzionamento. I due tool utilizzati non impiegano le API ufficiali rilasciate dai rispettivi social. Ciò avviene a causa della forte limitazione all'estrazione dati che viene imposta, nel caso specifico da Facebook e Instagram.

Questa limitazione rende necessario ricorrere a soluzioni che contrastano le linee guida definite dai fornitori dei servizi social, adottando tecniche in grado di eludere i controlli messi in campo e giungere all'obiettivo del progetto garantendo efficienza.

Le piattaforme social vengono continuamente sottoposte a nuovi sviluppi ed aggiornamenti, causando ovvi problemi all'operatività degli strumenti scelti. Ne consegue un continuo aggiornamento dei tool ed una attenta ricerca per la gestione tecnica del loro funzionamento.

[4] Uso improprio ed esterno dei dati.

[5] `https://pypi.org/project/facebook-scraper/`

[6] `https://instaloader.github.io/`

4.3 Soluzioni implementative per Facebook

4.3.1 Estrazione e formato dei dati

Il connettore di Facebook basa il suo funzionamento sul progetto "facebook-scraper". L'acquisizione e la gestione dei dati estratti da questo social network, sono state effettuate con l'obiettivo di essere rese fruibili da parte di sistemi di analisi dati (Big Data Analytics). Il tool opera direttamente effettuando richieste GET sulle diverse estensioni dei link, a partire da un target[7] dato in input. Nel caso specifico, il route[8] definito dall'API genera l'input per le singole funzioni dello scraper, avviando il processo di identificazione del target e conseguente iterazione sulla gerarchia di link collegati ad esso, ricavando i dati richiesti. Sono state definite in totale tre route o "cammini" per le API, rispettivamente per lo scraping di profili, pagine e gruppi.

Un esempio di route per l'estrazione di tutti i dati di un profilo è:

localhost:5000/profile/username/posts

I dettagli relativi alle singole funzioni sviluppate sono:

- **Profili**: in "profile.py" vengono descritte tutte le funzioni per lo scraping di profili pubblici e privati. In particolare si ha la possibilità di scaricare le informazioni generali di un profilo ed anche tutti i post, le foto ed i video, con relativi commenti e reazioni. La funzione "get_posts" di facebook-scraper restituisce un JSON contenente una moltitudine di informazioni da processare. Di conseguenza sono state sviluppate funzioni in grado di elaborare il file JSON ed estrarre ciò di interesse. Per le foto, osservato che Facebook prevede la presenza di foto in bassa qualità (identificate come anteprime) e foto in alta qualità (identificate come foto dei post) si è provveduto a differenziare il download di questi media in funzioni differenti. Si riporta in Appendice al paragrafo A.1.2 il codice per il download dei post e dei media da un profilo.

[7] Si fa riferimento all'obiettivo dell'attività di scraping.
[8] Si fa riferimento al percorso definito dal link relativo.

- **Pagine**: in "pages.py" vengono descritte le funzioni per lo scraping di pagine. Come descritto per i profili, anche in questo caso si ha la possibilità sia di estrarre le informazioni generali di una pagine, che i relativi post con media, commenti e reazioni.

- **Gruppi**: in "groups.py" sono definite le funzioni per lo scraping di gruppi pubblici e privati. Per i gruppi pubblici valgono le stesse modalità descritte per profili e pagine, mentre per i gruppi i privati lo scraping è limitato solo all'estrazioni di informazioni generali e delle indicazioni sugli amministratori. Lo scraping completo di un gruppo privato può essere effettuato solamente se l'account (ed i relativi cookies) utilizzato per lo scraping risulta appartente al gruppo.

Facebook-scraper è stato selezionato come migliore servizio da integrare nel funzionamento del connettore di dati dal social Facebook. Il confronto con gli altri tool validi individuati è riportato in Tabella 1.

I dati raccolti attraverso l'integrazione di questo tool sono stati sottoposti ad una precisa organizzazione come descritto nel paragrafo 4.5.1. L'insieme delle qualità descritte in Tabella 1 fanno si che facebook-scraper garantisca un'operatività continua nell'estrazione delle informazioni. In particolare la possibilità di gestione del tempo nelle richieste effettuate consente di simulare la normale attività di interazione umana con il servizio web, evitando il riconoscimento di automazione con conseguente ban. Gli ulteriori accorgimenti relativi alle tecniche di elusione dei controlli sono descritti al Paragrafo 3.2.

Tabella 1: Confronto Tool per Scraping di Facebook.

FACEBOOK-SCRAPER	**FACEBOOK-POST-SCRAPER-SELENIUM**	**FACEBOOK-SCRAPER-SELENIUM**
No API	No API	No API
Login non obbligatorio	Login obbligatorio	Login obbligatorio
Mantenimento di sessione di login	-	-
Gestione dei cookies sia in formato NETSCAPE che in formato JSON	-	-
Download anonimo completo di profili pubblici e privati	Download anonimo dei soli post di profili pubblici, privati	Download anonimo del solo testo dei post di profili pubblici e privati
Download di hashtag, commenti, tag geografici e descrizioni per ogni post	Download di commenti per ogni post (con foto e link)	-
Download di tutte le reazioni per ogni post	Download di solo tre reazioni per post	-
Download anonimo completo di pagine	Download anonimo dei post delle pagine	Download anonimo dei post delle pagine
Download anonimo di gruppi pubblici e privati	-	Download dei post dei gruppi pubblici e privati
Download di file JSON	Download di file JSON	Download di file CSV
I media vengono estratti tramite metodo GET	I media vengono estratti tramite metodo GET	-

Personalizzazione di filtri di ricerca	-	-
-	Utilizzo di “Selenium”	Utilizzo di “Selenium”
Download in elenco degli amici	-	-
Il download di profili privati avviene solo se l’account usato fa parte della lista degli amici dall’account target	Il download di profili privati avviene solo se l’account usato fa parte della lista degli amici dall’account target	Il download di profili privati avviene solo se l’account usato fa parte della lista degli amici dall’account target
Orientato al solo scraping di profili di terzi	Orientato al solo scraping di profili di terzi	Orientato al solo scraping di profili di terzi

4.3.2 Risoluzione del contrasto

Durante il funzionamento del tool, si incorre facilmente in errori dovuti alle soluzioni anti-scraping. In particolare si ricevono i seguenti codici di stato[9] e comunicazioni:

- **429 Too Many Requests**: il codice di stato 409 viene presentato quanto viene superato il limite di richieste da parte di un indirizzo IP in un determinato arco temporale. Una volta presente questo codice, sarà necessario riavviare il processo di scraping.

- **Temporary Ban**: errore presentato quando è stata individuata un’attività sospetta da parte di un account. Da test effettuati, il ban temporaneo di un profilo dura 24/48 ore, successivamente torna ad essere operativo.

- **Account disabled**: qualora un account sottoposto a un ban temporaneo continui ad effettuare richieste, viene presentato l’errore e la comunicazione di ban permanente a causa dell’attività non consentita. L’account non risulterà essere più recuperabile.

[9]Codici di stato HTTP, indicano l’esito di una comunicazione HTTP

- **Unusual Activity**: avviso di attività sospetta proveniente da un account. Viene bloccata l'attività in corso senza ulteriori "provvedimenti". Nel caso in cui lo stesso account, con lo stesso IP, continui operazioni di tipo massivo, si incorrerà in modo certo a un ban temporaneo.

Per limitare la comparsa degli errori sopra riportati è stato sviluppato un sistema di rotazione dei cookies, come presentato al paragrafo 3.2.3.

I cookie all'interno del progetto sono stati gestiti ed utilizzati per il login degli account utilizzati per lo scraping. La gestione si è basata sulla estrazione e memorizzazione dei valori necessari per il funzionamento dei tool scelti.

Per estrarre i cookie relativi alla sessione dei singoli account si è fatto uso di Selenium, grazie al quale, automatizzando le operazioni su browser Chromium in modalità headless, si sono potuti estrarre i valori dei cookie di interesse per il login.

Si riporta il codice relativo all'implementazione di Selenium ed un esempio dei valori dei cookie estratti in Appendice A.1.4.

La funzione "cookie_rotation" ha consentito il mantenimento di una continuità operativa del tool, grazie all'elusione dei controlli. Si riporta il codice della funzione:

```
def rotatecookie(self, read_from_file= True):
                if read_from_file:
                        with open('accounts.json') as f:
                                accounts = json.load(f)
                count=0
                for account in accounts:
                        count+=1
                num=randrange(count)
                print("using cookies num", num+1)
        return set_cookies('./cookies/cookie'+str(num+1)+'.json')
```

Oltre alla soluzioni adottata sui cookie, si è prevista per ogni azione di download con richieste dirette al server target, l'aggiunta di tempi d'attesa nell'ordine dei secondi. Rallentando l'attività si è aumentata la resistenza del tool contro eventuali

azioni di ban, immedesimando l'utilizzo del social da parte di un umano, inteso come utente reale e non bot.

```
time.sleep(15)
```

La funzione "time.sleep" in Python, sospende l'esecuzione di un determinato processo sulla base del numero di secondi inseriti in input.

Tutte le funzioni elencate necessitano di login con account validi. La gestione degli account avviene attraverso la memorizzazione delle credenziali in file JSON, dal quale vengono estratte per avviare i processi di login.

4.3.3 Aggiornamenti

A causa dell'attivo e consistente contrasto ed all'aggiornamento tecnico delle infrastutture dei social, durante lo sviluppo si sono presentate diverse problematiche relative al corretto funzionamento dello scraping. Ogni questione è stata prontamente risolta grazie al rilascio di nuove versioni del tool. La cronistoria dei rilasci è presente nello schema riportato al paragrafo 5.1.1.

4.4 Soluzioni implementative per Instagram

4.4.1 Estrazione e formato dei dati

Il connettore di Instagram basa il suo funzionamento sul progetto "instaloader". Instaloader consente il download di profili completi di media (foto e video) e metadati[10] per ogni post. L'acquisizione e la gestione dei dati estratti da questo social network, sono state effettuate con l'obiettivo di essere rese fruibili da parte di sistemi di analisi dati (Big Data Analytics).

Instaloader è stato selezionato dopo la comparazione del suo funzionamento con altri tool open source individuati. Il confronto è riportato in Tabella 2. L'attività effettuata con instaloader ha subito una organizzazione e gestione simile a

[10]Serie di informazioni aggiuntive sui dati.

quella di facebook-scraper. Le principali differenze, riguardanti l'approccio di funzionamento del tool, sono state uniformate con particolare attenzione alla gestione dei dati in output.

Lo sviluppo dell'API prevede la presenza di route, le quali generano l'input per le singole funzioni dello scraper, avviando l'iterazione sulla gerarchia di link collegati al target, ricavando i dati richiesti. Un esempio di route per l'estrazione dei post con il connettore di Instagram è:

```
localhost:5000/<username>
```

Anche nel caso di Instagram, le attività messe in campo per l'elusione dei controlli anti-scraping sono descritte al Paragrafo 3.2.

Le principali funzioni sviluppate ed impiegate per lo scraping di questo social sono:

- **Post, foto e video**: la funzione "profile_dl" consente di estrarre tutti i post di un profilo pubblico senza effettuare il login e tutti i post di un profilo privato se si effettua il login e se l'account in uso fa parte dei follower approvati dal target. È possibile anche effettuare il download dei post in cui l'obiettivo è stato taggato. La funzione prevede un controllo della frequenza delle richieste, limitando il numero delle stesse ed aggiungendo ritardo nell'ordine dei secondi. Insieme ai media relativi a tutti i post viene memorizzata anche la foto profilo dell'account. Per ogni post estratto, viene memorizzato anche un file JSON contenente informazioni aggiuntive come i commenti, il numero di mi piace, l'eventuale posizione GPS, la data e l'ora di pubblicazione, le informazioni del profilo, eventuale connessione del profilo con un account Facebook ed altri metadati.

- **Storie**: la funzione "stories_dl" consente di estrarre tutte le storie valide[11] pubblicate da un utente. La funzione necessita di login al social e prevede il controllo della frequenza.

[11]Per validità di una storia si fa riferimento, secondo le policy di Instagram, al contenuto pubblicato entro le 24 ore dalla visione.

- **Followers e seguiti**: le funzioni "followers_dl" e "followee_dl" prevedono la possibilità di restituire in output elenchi in JSON dei follower e dei seguiti dell'account target. Necessitano di login e sono molto vulnerabili alle azioni di contrasto. La loro vulnerabilità nasce dal numero di elementi da estrarre. Nella maggioranza dei casi, il numero di follower o di seguiti è maggiore rispetto al numero di post pubblicati ed appartenente all'ordine delle centinaia o migliaia. Ne consegue che il numero di richieste segue il numero di elementi, esponendo l'azione al controllo.

- **Geolocalizzazione**: la funzione "location_dl" consente, a partire dall'input di un luogo, di effettuare l'estrazione di tutti i post geolocalizzati in quel punto. Ne consegue che questa funzione rappresenta un'estensione rispetto al normale download dei post da un profilo, in quanto estende la possibilità di ricerca sulla base della posizione.

Il tool offre la possibilità di analizzare il comportamento storico di un utente, confrontando i dati già scaricati e memorizzati rispetto ad una nuova azione di scraping. Ciò oltre ad evitare ridondanza, consente anche di ottimizzare le richieste al server, riducendone il numero.

Anche in questo caso, tutte le funzioni elencate che necessitano di login con account validi, gestiscono le credenziali attraverso file JSON, dal quale vengono estratte per avviare i processi.

Tutti i media vengono memorizzati nella qualità massima, ovvero la stessa presente alla normale fruizione dei dati dal social. Si riportano in Appendice al Paragrafo A.2.1 le funzioni di scraping adottate per lo sviluppo del connettore.

Tabella 2: Confronto Tool per Scraping di Instagram.

INSTALOADER	**INSTAGRAPI**	**INSTAGRAM-SCRAPER-2021-SELENIUM**
No API	Basato su API ufficiale di Instagram	No API
Login obbligatorio per alcune azioni	Login obbligatorio	Login obbligatorio
Supporto autenticazione a due fattori	Supporto autenticazione a due fattori	Supporto autenticazione a due fattori
Download anonimo di profili pubblici e privati	Download anonimo di profili pubblici, privati	Download anonimo di profili pubblici, privati
Il download di profili privati avviene solo se l'account usato fa parte della lista dei follower dell'account target	Il download di profili privati avviene solo se l'account usato fa parte della lista dei follower dell'account target	Il download di profili privati avviene solo se l'account usato fa parte della lista dei follower dell'account target
Download di storie, mi piace, commenti e tag geografici	Download di storie, mi piace, commenti e tag geografici	-
Download diretto di foto e video	-	-
Download dei file in JPG, MP4 e JSON	Download di file JSON	Download di file JSON
-	-	Utilizzo di Selenium
Rilevazione di cambio di username	-	-
Rilevazione dei post eliminati	-	-

Download di metadati aggiuntivi	-	-
Personalizzazione di filtri di ricerca	-	-
Mantenimento di sessione di login	-	-
Permette il download di post appartenenti ad uno specifico periodo temporale	-	-
Rilevazione automatica e continuazione di download interrotti	-	-
-	Challenge resolver (per verification code login su email e sms)	-
Presenza di buon supporto da parte della community	-	-

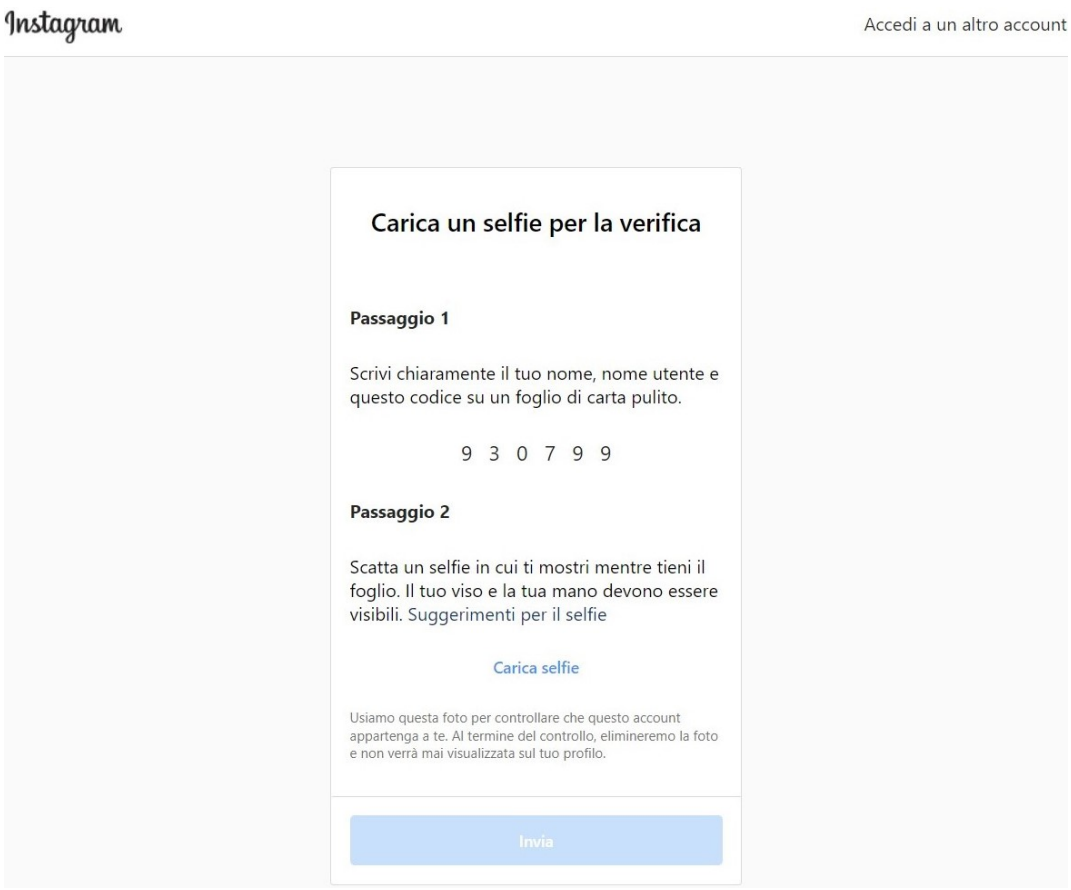
4.4.2 Risoluzione del contrasto

Come per il tool di Facebook, anche per lo scraping di Instagram, si presentano errori dovuti alle soluzioni anti-scraping:

- **429 Too Many Requests**: il codice di stato 429 viene presentato quanto viene superato il limite di richieste da parte di un indirizzo IP in un determinato arco temporale. Una volta presente questo codice, sarà necessario riavviare il processo di scraping.
- **401 Unauthorized**: il codice 401 viene presentato quando si tenta di effettuare il download di dati da profili privati o senza aver effettuato il login al servizio.

- **Ban**: diversamente da quanto accade con Facebook in cui è presente anche un soft-ban, per Instagram il ban di un profilo risulta essere definitivo in caso di azioni non consentite. Ciò avviene con più frequenza quando ad le azioni non riconosciute sono effettuate da un account appena creato. Lo studio e lo sviluppo del connettore ha evidenziato come Instagram attui una politica di controllo nei confronti di account appena creati, attenzionando e limitando il numero di operazioni consentite.

- **Riconoscimento "umano" anti-bot**: una volta che un account è stato segnalato a causa di azioni sospette, viene richiesta l'identificazione personale del soggetto per eliminare eventuali bot. Nel caso in cui ciò non avvenisse, il profilo verrà sottoposto a ban permanente. Si riporta schermata di richiesta fotografia "selfie" con specifiche azioni da effettuare per il riconoscimento dell'utente in qualità di persona fisica proprietaria dell'account.



Lo studio e lo sviluppo di tecniche di elusione dei controlli e del contrasto sopra descritto si è attuato adottando le stesse idee implementative del connettore di Facebook.

In questo caso, come fatto per la funzione "cookie_rotation", osservando la diversa gestione dei cookie da parte del tool per il connettore di Instagram, si è sviluppata una funzione di "account_rotation" di cui si riporta il codice.

```
class account:
    def rotate_accounts(read_from_file= True):
        if read_from_file:
            with open('accounts.json') as f:
                accounts = json.load(f)
                count=0
            for account in accounts:
                count+=1
            num=randrange(count+1)
            print('Using account num:',num+1)
            user= accounts[num]['user']
            passw= accounts[num]['pass']
            return user, passw
```

4.4.3 Aggiornamenti

Come descritto per il connettore di Facebook, le problematiche di sviluppo del connettore e di impiego del tool instaloader sono state causate, oltre che dalle soluzioni anti-scraping, dagli aggiornamenti della piattaforma. Le maggiori difficoltà affrontate nello sviluppo hanno riguardato la gestione del login al servizio, continuamente aggiornata da Instagram sia in termini di funzionalità che in termini di sicurezza.

Al Paragrafo 5.1.1 viene presentata la cronistoria degli aggiornamenti del tool su cui si basa il connettore.

4.5 Tecnologie utilizzate

Le soluzioni tecniche adottate per lo sviluppo di questo progetto sono state basate sul concetto di portabilità e operatività continua del servizio di raccolta dati.

Il linguaggio scelto per lo sviluppo del progetto è stato Python[12], il quale grazie alla vasta offerta di librerie e strumenti dedicati, è risultato essere idoneo ad applicazioni di Web Scraping e di analisi ed elaborazione dati. Inoltre Python si presenta come uno strumento vantaggioso per lo sviluppo di API[13], fondamentali per permettere il riuso del codice e garantire un utilizzo e un'inclusione delle funzionalità da parte di altri sviluppatori.

Le scelte tecnologiche adottate sono state definite tramite l'impiego di strumenti Open Source, di formati dati universali e leggibili e di paradigmi architetturali per lo sviluppo web.

Lo sviluppo e l'impiego di software container, ha permesso di massimizzare la portabilità del servizio, garantendo uniformità e produttività.

L'individuazione di tool open source si è basata su uno studio e confronto delle carattistiche, ponendo attenzione anche all'attività e al seguito delle community di sviluppatori, in modo tale da garantire continuità e costanza nell'aggiornamento dei servizi.

4.5.1 Formato dei dati

Lo standard individuato per la gestione dei dati estratti è il formato JSON.

JSON è un formato di testo per la serializzazione di dati strutturati basato sul linguaggio di programmazione JavaScript, dal quale però non dipende. JSON rappresenta quattro tipi di dato fondamentali (stringhe, numeri, valori booleani e nulli) e due tipi strutturati (oggetti e vettori).[21] Grazie alla sua chiara sintassi, rappresenta uno standard interpretabile facilmente sia da persone che da macchine garantendo portabilità e interoperabilità multipiattaforma, fondamentale per l'impiego delle API sviluppate in questo progetto.

[12] `https://www.python.org/`

[13] Application Programming Interface.

L'alta leggibilità del formato JSON ha permesso una manipolazione dei file di output, garantendo soluzioni tecniche idonee per la successiva estrazione dei dati.

La successiva soluzione per il salvataggio dei dati richiesti si basa su formati standard di file multimediali come MP4 e JPG. Essendo lo scraping massivo di dati e la conseguente ingestione in sistemi di analisi dati l'obiettivo finale del progetto, si è reso necessario comprimere tutti i dati raccolti in formato ZIP.

Il formato ZIP [22] è un formato di compressione dati senza perdita di informazioni in grado di ridurre lo spazio occupato dai file. L'utilità della compressione dei file si rende fondamentale nel processo di ingestione dei dati in sistemi di Big Data Analytics.

4.5.2 API

Il progetto si è basato sullo sviluppo di API RESTful utilizzando il servizio Flask[14].

Flask è un framework per applicazioni web, caratterizzato dalla sua leggerezza ed ampia compatibilità. Offre scalabilità e gestione di progetti complessi, non richiedendo specifici strumenti o scelte progettuali; in questo modo vengono favorite le opzioni implementative delineate dello sviluppatore. Le funzionalità di Flask impiegano due dipendenze: Werkzeug[15] e Jinja2[16]. Werkzeug gestisce le attività di routing, debugging ed il protocollo di trasmissione WSGI[17], mentre Jinja2 gestisce i template per le API. Lo sviluppo di API REST tramite Flask consente di instanziare le singole applicazioni, associando singoli route (percorsi) alle varie funzioni, sotto forma di URL. La risposta alle varie richieste, fornita dal server, fornisce in aggiunta un HTTP Response (codice) che, a seconda del caso, individua un successo o un errore di client o server.[23]

Si riporta un estratto di codice del connettore di Facebook inerente all'utilizzo di Flask in Appendice A.1.1.

[14] `https://flask.palletsprojects.com/en/2.2.x`
[15] `https://werkzeug.palletsprojects.com/en/2.2.x`
[16] `https://jinja.palletsprojects.com/en/3.1.x`
[17] Web Server Gateway Interface.

4.5.3 Docker

Entrambi i connettori sono stati posti in container utilizzando Docker. Come introdotto al Paragrafo 3.1.5, per la creazione dell'immagine occore generare il rispettivo Dockerfile. In appendice si riporta a titolo d'esempio il Dockerfile per il connettore di Facebook (Paragrafo A.1.3).

4.5.4 Strumenti di automazione

L'automazione del software viene impiegata nell'ambito del testing dei servizi e dell'interazione automatica con elementi dell'interfaccia utente. In questo progetto, è stato individuato Selenium come strumento open source in grado di effettuare operazioni automatiche sui browser.

Selenium[18] consente di automatizzare delle operazioni su browser web, consentendo anche l'estrazione di informazioni visive.[24] L'impiego di Selenium nel progetto è consistito nell'accesso automatico ai siti web di interesse e nell'estrazione dei valori relativi ai cookies persistenti (descritti al paragrafo 3.1.6).

Il funzionamento di Selenium è basato sull'interpretazione e la creazione di richieste HTTP per ogni comando scritto dallo sviluppatore. Ogni comando viene successivamente inviato al driver del browser individuato come ambiente di testing e di lavoro. Sono supportati i maggiori browser quali Chrome/Chromium, Firefox, Edge, Explorer e Safari. Per le scelte progettuali, osservato l'impiego di container basati su Linux, l'automatizzazione è stata implementata con driver Chromium[19].

Grazie alla possibilità di impostare il WebDriver in modalità headless[20], sono stati recuperati automaticamente i dati aggiornati dei singoli account da impiegare, garantendo operatività continua.

[18] `https://www.selenium.dev/`

[19] `https://www.chromium.org/chromium-projects/`

[20] Headless Browser, assenza di interfaccia grafica.

Capitolo 5

Test

5.1 Prestazioni del Software

Nel seguente capitolo vengono presentati dati e grafici inerenti al funzionamento e l'efficienza del software prodotto, facendo riferimento anche ai tool open source impiegati.

5.1.1 Confronti sui tool

Durante lo studio e lo sviluppo delle soluzioni software descritte nei precedenti capitoli, si è potuto notare come l'attività di scraping nei confronti di Facebook ha subito meno variazioni nel funzionamento rispetto a Instagram. Ciò può essere osservato confrontando il rilascio di nuovi versioni dei tool integrati nei connettori, effettuati a causa di presenza di problemi per aggiornamenti delle piattaforme. Si riporta uno schema di confronto delle versioni rilasciate a partire da Luglio 2022 (inizio del progetto di ricerca e sviluppo della tesi) di "instaloader"[1] e di "facebook-scraper"[2] .

[1] `https://pypi.org/project/instaloader/#history`

[2] `https://pypi.org/project/facebook-scraper/#history`

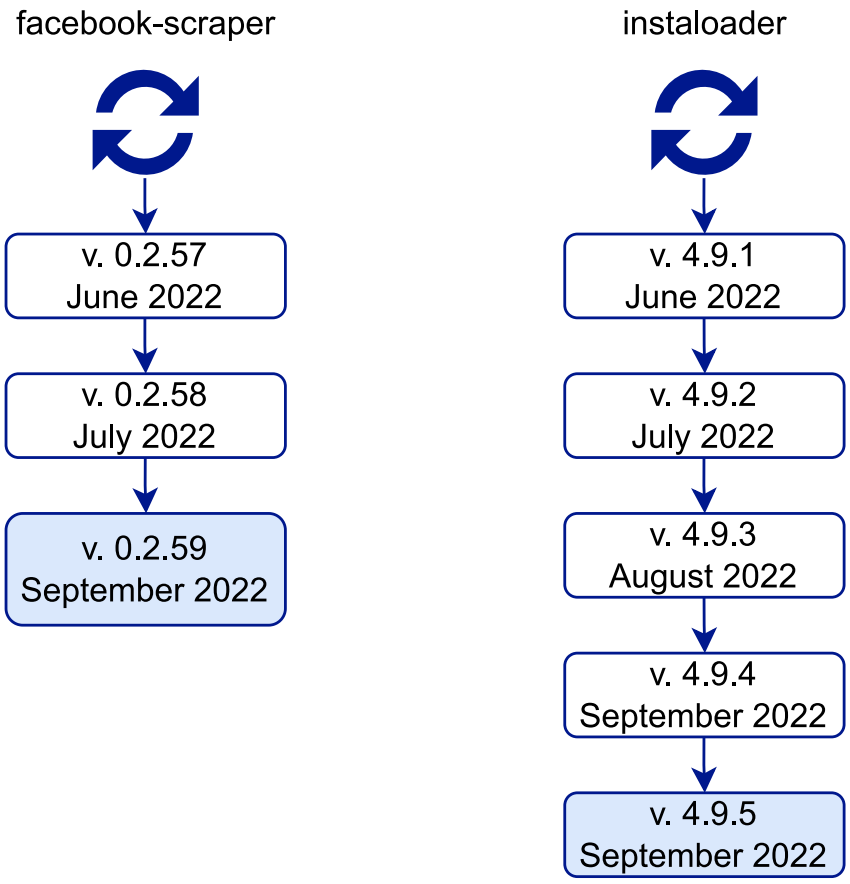


Figura 10: Storia degli aggiornamenti dei tool integrati

5.1.2 Confronti sui connettori

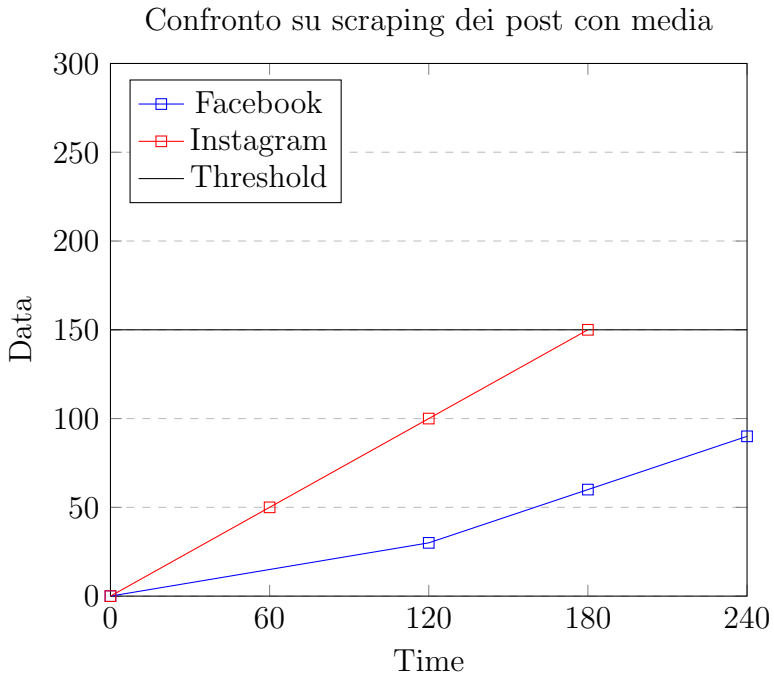
L'operatività dei due connettori sviluppati si differenzia in base a tempistiche di scraping e quantità di dati recuperati. In seguito a molteplici prove e allo studio delle risposte di contrasto da parte dei server, si sono potute identificare diverse soglie massime di azione continuativa prima di incorrere nei vari provvedimenti anti-estrazione. Nel grafico di seguito riportato si possono osservare i diversi comportamenti dei connettori nell'azione di scraping di post di profili, a fronte delle soluzioni e caratteristiche sviluppate:

- **Tempo e delay imposto**: per entrambi i connettori è stato previsto un ritardo nelle operazioni di 15 secondi. I tempi riportati nei grafici includono

il tempo aggiuntivo di ritardo. Per il connettore di Facebook, non viene preso in considerazione l'arco temporale necessario all'estrazione dei cookie ed al login negli account.

- **Treshold**: per entrambi i connettori sono state previste delle soglie massime di dati da estrarre.

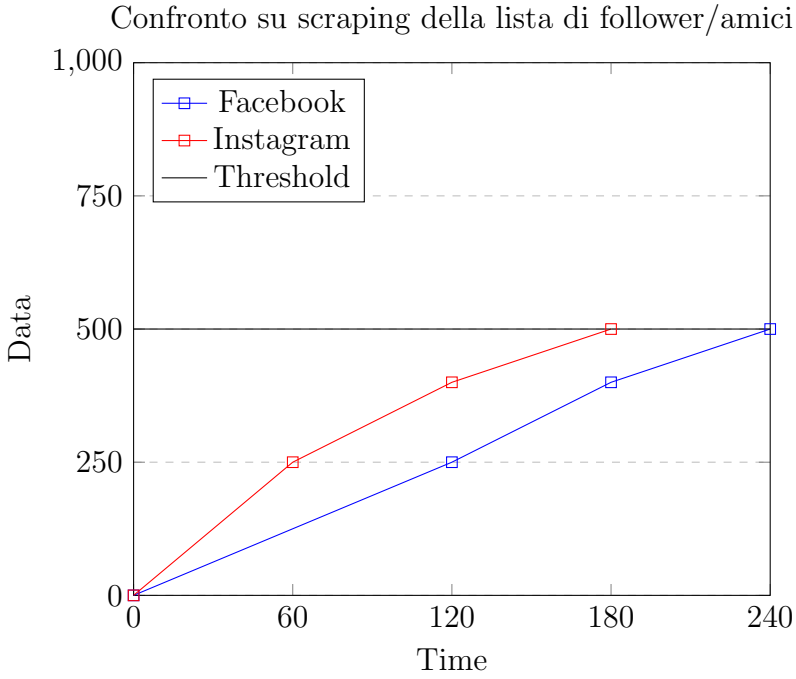
- **Connessione e memoria**: per i test è stata impiegata una connessione avente una media in download di 100 Mbps.



La soglia impostata equivale a 150 post.

Nel caso in cui venga raggiunta la soglia si procede con l'interruzione del servizio e l'avvio della funzione cookie o account rotation.

La media dello spazio di memoria occupato da una memorizzazione di 100 post (con foto e metadati in JSON) è di 100 Mb.



La soglia impostata equivale a 500 follower/amici.

Anche in questo caso, una volta raggiunta la soglia si interrompe il servizio per poi riprenderlo a seguito della rotazione degli account.

La liste prodotte in output sono dei file JSON e la loro occupazione di memoria dipende direttamente dalla quantità di dati memorizzati. Il file JSON può arrivare ad occupare al massimo pochi Mb di spazio su disco.

5.1.3 Osservazioni

Il confronto dei connettori mette in luce come lo scraping di Instagram sia circa il doppio più veloce rispetto a quello di Facebook. La motivazione risiede nella gestione dei dati e nella quantità di informazioni aggregate ad un singolo post o ad una singola richiesta al server.

I principali problemi riscontrati nell'operatività del software sono causati dalle soluzioni anti-scraping.

In particolare, da quanto osservato, il fulcro del contrasto avviene tramite l'analisi delle richieste, del numero di elementi estratti e dall'account utilizzato.

Capitolo 6

Conclusioni

Con la fase di test dei connettori e il raggiungimento dell'operatività massima si conclude questo progetto di tesi.

Il maggiore sforzo nello studio e sviluppo di questo progetto è consistito nella gestione e nella ricerca di soluzioni valide per l'elusione dei controlli imposti dai servizi a discapito dell'attività di scraping.

La costanza degli aggiornamenti e l'introduzione delle nuove funzionalità di opposizione all'estrazione introdotte da Facebook ed Instagram hanno fatto sì che il lavoro di tesi richiedesse una continua attenzione e frequenza nel testing del software, causando molteplici cambiamenti nell'approccio di sviluppo e ricercando nuove soluzioni.

Le funzioni implementate rappresentano ottimi compromessi tra operatività continua ed efficienza. L'alto numero di dati raccolti e gestiti rappresenta un input ottimale per la successiva ingestione in sistemi di Big Data Analytics.

Il mantenimento nel lungo periodo di un software stabile e facile da aggiornare rappresenta l'obiettivo finale del progetto, raggiunto grazie alle soluzioni studiate e sviluppate in questo lavoro di tesi.

Si fa presente che tutti i dati nell'ambito dello sviluppo e del test del progetto sono stati trattati ai fini di ricerca scientifica.

6.1 Sviluppi futuri

I principali aspetti che possono rappresentare un futuro sviluppo del progetto possono essere i seguenti:

- la possibilità di integrazione del software sviluppato nell'ambito dei sistemi distribuiti, come proposto al Paragrafo 3.2.7. Questa idea implementativa rientra nelle possibilità di elusione dei controlli imposti dai fornitori dei servizi social;
- gestione sicura degli account da impiegare tramite la memorizzazione delle credenziali crittografate. Un esempio può essere prevedere l'utilizzo di SQLite con SEE[1];
- la creazione di un generatore automatico di account, in modo da non impiegare attivamente l'operatore nel fornire utenze manualmente;
- implementare ip-rotation proposta al Paragrafo 3.2.6 ed eventuali Proxy Server.

Il progetto, osservato il continuo cambiamento delle piattaforme, necessita di aggiornamento costante. Le funzioni sviluppate sono state sviluppate in modo da essere facilmente adattate ad eventuali novità tecniche introdotte dai social, resta però comunque l'esigenza di un continuo sviluppo, soprattutto per rafforzare l'elusione dei sistemi anti-scraping.

[1] SQLite Encryption Extension, `https://sqlite.org/com/see.html`

Appendice A

Estratti di codice

In questa appendice sono inserite parti di codice del progetto di tesi. Per motivi di privacy, di formattazione e di grafica sono stati rimossi e modificati alcuni elementi dai file riportati.

A.1 Connettore di Facebook

A.1.1 Flask API

Si riporta il codice dal "main" del progetto per Facebook, dove si instanzia Flak e si genera la API con le conseguenti "route" (esempio da "profile.py") per raggiungere le varie risorse.

```
from facebook_scraper import *
from flask import Flask
from flask_restx import Api
from fbpages import api as pages
from fbprofile import api as profile
from fbgroups import api as groups

app = Flask(__name__)
api = Api(app)
```

```
api.add_namespace(profile)
api.add_namespace(pages)
api.add_namespace(groups)

if __name__ == "__main__":
    app.run(host="127.0.0.1", port=5000)
```

```
#profile.py
from flask_restx import Namespace, Resource
from flask import request

api= Namespace('profile')

@api.route('/<username>/info')
class ProfileInfo(Resource):
   def get(profileinfo, username):
        ...
```

A.1.2 Funzioni di scraping

Scraping di profili e download dei relativi media.

```
@api.route('/<username>/posts')
#download user's post
class UserPost(Resource):
    def get(UserPost,username):
         name=username
         posts = dl_posts(name)
         print(posts)
         Path(name).mkdir(parents=True, exist_ok=True)
         usernamepost= name+"_posts.json"
         with open(name+"/"+usernamepost, "w") as f:
```

```
            f.write(json.dumps(posts, default=str,indent=4))
        download().dl_images(quality=download.LOW_QUALITY_KEY,
        posts=posts,name=name)
        download().dl_images(quality=download.HIGH_QUALITY_KEY,
        posts=posts,name=name)
        download().dl_video(name=name, posts=posts)
        download().create_archive()
        return json.dumps(posts, default=str, indent=4)
```

A.1.3 Docker

Si riporta il Dockerfile per la generazione dell'immagine Docker.

```
FROM python:3.10

WORKDIR /instagram-scraping
COPY requirements.txt requirements.txt
RUN pip3 install -r requirements.txt

COPY . .

CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

A.1.4 Cookie

Si riporta il codice relativo all'impiego di Selenium per l'estrazione dei cookie dal login in Facebook.

```
def create_cookie(self,email,passw,count):


    chrome_options = webdriver.ChromeOptions()
    prefs={"profile.default_content_setting_values.notifications":
    2}
```

```
    chrome_options.add_experimental_option("prefs",prefs)
    chrome_options.headless = True
    chrome_options.add_argument("--log-level=3")
    driver = webdriver.Chrome(ChromeDriverManager
    (chrome_type=ChromeType.CHROMIUM).install(),
    chrome_options=chrome_options)
    driver.get("http://www.facebook.com")
    WebDriverWait(driver, 10).until(EC.element_to_be_clickable(
    (By.CSS_SELECTOR, 'button[data-cookiebanner="accept_button"]'))
    ).click()
    username = WebDriverWait(driver, 10).until
    (EC.element_to_be_clickable(
    (By.CSS_SELECTOR, "input[name='email']")))
    password = WebDriverWait(driver, 10).until
    (EC.element_to_be_clickable(
    (By.CSS_SELECTOR, "input[name='pass']")))
    username.clear()
    username.send_keys(email)
    password.clear()
    password.send_keys(passw)

    WebDriverWait(driver, 2).until
    (EC.element_to_be_clickable(
    (By.CSS_SELECTOR, "button[type='submit']"))).click()
    cookies_list=[]
    cookies_list.append(driver.get_cookie('c_user'))
    cookies_list.append(driver.get_cookie('xs'))
    filename= "./cookies/cookie"+str(count)+".json"
    with open(filename, "w") as f:
        f.write(json.dumps(cookies_list, default=str, indent=4))
    return (json.dumps(cookies_list, default=str, indent=4))
```

L'output in JSON dei valori estratti dai cookie da Selenium all'atto del login è composto da due valori necessari per il funzionamento del tool: il valore denominato "c_user" rappresenta l'identificativo univoco dell'account, mentre il valore "xs" rappresenta l'identificativo univoco della sessione di login.


```
[
    {
        "domain": ".facebook.com",
        "expiry": 1697301133,
        "httpOnly": false,
        "name": "c_user",
        "path": "/",
        "sameSite": "None",
        "secure": true,
        "value": "123456789"
    },
    {
        "domain": ".facebook.com",
        "expiry": 1697301133,
        "httpOnly": true,
        "name": "xs",
        "path": "/",
        "sameSite": "None",
        "secure": true,
        "value": "njdbjbvbjabfiu324%%638914uh'?rtf"
    }
]
```


A.1.5 Output in JSON

Si riporta un esempio di output in JSON dei dati estratti da una richiesta di scraping per informazioni generali di un profilo.

```
{
    "Friend_count": 821,
    "Follower_count": null,
    "Following_count": 292,
    "cover_photo_text": "Cover Photo: Paolo Verdi's photo.",
    "cover_photo": "https://scontent.fmxp7-1.fna.fbcdn.net/",
    "profile_picture": "https://scontent.fmxp7-1.fna.fbcdn.net/",
    "id": "100003940579155",
    "Name": "Paolo Verdi",
    "Education": {
        "University": "Dipartimento di informatica ",
        "High school": "Liceo Scientifico"
    },
    "Places lived": [
        {
            "link": "//profile.php?id=123456",
            "text": "Roma",
            "type": "Current city"
        },
        {
            "link": "/profile.php?id=123456",
            "text": "Roma",
            "type": "Hometown"
        }
    ],
    "Contact info": "/paoloverdi\nFacebook",
    "Basic info": "Male\nGender",
    "Life events": "",
    "Friends": [
        {
            "id": 100000000000006,
```

```
            "link": "/username",
            "name": "Luigi Bianchi",
            "profile_picture": "https://...",
            "tagline": "Milan, Italy"
        },
        {
            "id": 100000000000000,
            "link": "/username",
            "name": "Mario Rossi",
            "profile_picture": "https://...",
            "tagline": "Rome, Italy"
        },
        {...}
    ]
}
```

A.2 Connettore di Instagram

A.2.1 Funzioni di scraping

Scraping di profili e e download dei relativi media.

```
class profile_dl(Resource):
    def get(self, username):
        L=login.login()
        MyRateController(instaloader.RateController)
        id = str(uuid.uuid4())
        posts = instaloader.Profile.from_username(L.context,
        username).get_posts()
        for post in posts:
            print(post.date)
            L.dirname_pattern = './'+id+'/posts'
```

```
            L.download_post(post, username)
        shutil.make_archive(id, 'zip', id)
        shutil.rmtree(id)

class stories_dl(Resource):
    def get(self, username):
        L=login.login()
        id = str(uuid.uuid4())
        MyRateController(instaloader.RateController)
        choices = {'stories': True,
                'posts': False,
                'profile_pic': True,
                'raise_errors': True}
        profile = instaloader.Profile.from_username(L.context,
        username)
        profiles = {profile}
        L.save_metadata = False
        L.dirname_pattern = './'+id+'/stories'
        L.compress_json = False
        L.post_metadata_txt_pattern = ''
        L.storyitem_metadata_txt_pattern = ''
        L.download_profiles(profiles, **choices)
        shutil.make_archive(id, 'zip', id)
        shutil.rmtree(id)class profile_dl(Resource):
```

A.2.2 Output in JSON

Si riporta un esempio di file JSON estratto da un profilo ed inerente ad un singolo post.

```
{
"node":{
    "typename": "GraphSidecar",
    "id": "1234567890",
    "shortcode": "ChPENRkoG01",
    "dimensions":{},
    "display _url":"https://scontent-seal-1.cchinstagram.com",
    "edge media _to tagged _user":{},
    "fact_check_overall_rating":null,
    "fact_check _information":null,
    "eating. info" :null,
    "sharing friction_info": {},
    "media _overlay _info":null,
    "media _preview":null,
    "owner"
    "is video":false,
    "has _upcoming_event":false,
    "accessibility_caption": "Photo by Luigi Verdi on August 14,
        2022.
        May be a closeup of 1 person.",
    "edge media _to caption":{},
    "edge media _to_comment": {},
    "comments_disabled": false,
    "taken_at_timestamp": 1660470411,
    "edge _1iked_by":{},
    "count": 226342,
    "edge media _preview_like":{},
    "location": null,
    "nft_asset_info":null,
    "thumbnail src":"https://scontent-seal-1.cdninstagram.com/",
    "thumbnail resources": [
```

```
    ],
    "coauthor_producers": [
    ], "pinned_for_users": [
    ],
    "edge_sidecar_to_children":{},
    "iphone_struct": {},
    "instaloader": {}
    }
}
```

Bibliografia

[1] Ryan Mitchell. *Web scraping with Python: Collecting more data from the modern web.* " O'Reilly Media, Inc.", 2018.

[2] Christopher Olston, Marc Najork, et al. Web crawling. *Foundations and Trends® in Information Retrieval*, 4(3):175–246, 2010.

[3] Jim Isaak and Mina J Hanna. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer*, 51(8):56–59, 2018.

[4] N. Tirino. *Cambridge analytica: il potere segreto, la gestione del consenso e la fine della propaganda.* Libellula edizioni, 2019.

[5] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *2013 IEEE Symposium on Security and Privacy*, pages 541–555. IEEE, 2013.

[6] Emilio Guida. *Intelligence, costante storica, variabile teorica e prospettive post-bipolari.* "Ledizioni LediPublishing", 2016.

[7] Maurizio Tesconi. Big data & social media intelligence. *Gnosis, Agenzia Informazioni e Sicurezza Interna*, 2017.

[8] Ian Lesser, John Arquilla, Bruce Hoffman, David F Ronfeldt, and Michele Zanini. *Countering the new terrorism.* RAND corporation, 1999.

[9] Vlad Krotov and Leiser Silva. Legality and ethics of web scraping. *Twenty-fourth Americas Conference on Information Systems*, 2018.

[10] Unione Europea. General data protection regulation, 2016.

[11] Lei Xu, Chunxiao Jiang, Jian Wang, Jian Yuan, and Yong Ren. Information security in big data: privacy and data mining. *Ieee Access*, 2:1149–1176, 2014.

[12] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.

[13] M. Masse. *REST API Design Rulebook*. Oreilly and Associate Series. O'Reilly Media, 2011.

[14] Madiha H Syed and Eduardo B Fernandez. *The software container pattern*. Association for Computing Machinery, New York, United States, 2015.

[15] Amit M Potdar, DG Narayan, Shivaraj Kengond, and Mohammed Moin Mulla. Performance evaluation of docker container and virtual machine. *Procedia Computer Science*, 171:1419–1428, 2020.

[16] David Kristol and Lou Montulli. Http state management mechanism. Technical report, RFC 6265, 2000.

[17] John Meehan, Cansu Aslantas, Stan Zdonik, Nesime Tatbul, and Jiang Du. Data ingestion for the connected world. In *CIDR*, 2017.

[18] Philip Russom et al. Big data analytics. *TDWI best practices report, fourth quarter*, 19(4):1–34, 2011.

[19] Andrew S Tanenbaum and Maarten Van Steen. *Sistemi distribuiti. Principi e paradigmi*. Pearson Italia Spa, 2007.

[20] Moreno Mancosu and Federico Vegetti. What you can scrape and what is right to scrape: A proposal for a tool to collect public facebook data. *Social Media + Society*, 6(3), 2020.

[21] Tim Bray. The javascript object notation (json) data interchange format. Technical report, RFC 7159, 2014.

[22] Paul Lindner. Registration of a new mime content-type/subtype, zip. Technical report, IANA, 1993.

[23] Miguel Grinberg. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[24] Unmesh Gundecha. *Selenium Testing Tools Cookbook*. Packt Publishing Ltd, 2015.

Ringraziamenti

A conclusione dell'elaborato desidero ringraziare anzitutto il Prof. Marco Anisetti per avermi accolto e seguito come suo studente per lo sviluppo di questa tesi, avente come oggetto temi inerenti ai suoi campi di ricerca.

Ringrazio anche il Dott. Antongiacomo Polimeno per avermi garantito una presenza costante e aiuto tecnico nello sviluppo del progetto, mettendo a disposizione la sua professionalità e competenza.

Infine ringrazio la mia famiglia e tutti i miei amici per il loro supporto in questi tre anni.