UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale Ingegneria Informatica e dell'Automazione

Metodologie di Deep Learning per individuare le tendenze nel settore fashion.

Deep Learning methodologies to identify trends in the fashion sector.

Relatore:

Prof. Emanuele Frontoni

Candidato:

Vagnoni Cristiano

Correlatore:

Prof. Marina Paolanti

A.A. **2020 /2021**

UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale Ingegneria Informatica e dell'Automazione

Metodologie di Deep Learning per individuare le tendenze nel settore fashion.

Deep Learning methodologies to identify trends in the fashion sector.

Relatore:

Prof. Emanuele Frontoni

Candidato:

Vagnoni Cristiano

Correlatore:

Prof. Marina Paolanti

A.A. **2020 /2021**

Università Politecnica delle Marche Facoltà di Ingegneria Corso di Laurea in Ingegneria Informatica e dell'Automazione Via Brecce Bianche – 60131 Ancona (AN), Italy

Instead of reality being passively recorded by the brain, it is actively constructed by it

David Eagleman

Indice

Elenco delle figure

Elenco dei listati

1. Introduzione

1.1 Descrizione del contesto

Apparire sui social network sembra essere un desiderio spinto da un bisogno associato all'approvazione sociale, al voler essere accettati e spalleggiati dagli altri. Per questo motivo, secondo questo studio, la nostra vita sociale è perlopiù vincolata alle piattaforme social offerte da internet. Non deve sorprendere dunque l'utilizzo che facciamo di questo strumento per lanciare messaggi alle persone che fanno parte della nostra cerchia.

Per riassumere, siamo strettamente connessi a internet e ai social network; essi fanno parte della nostra quotidianità. Così come fanno parte della nostra routine quotidiana concetti come "postare" o "farsi un selfie".

In un contesto aziendale e di marketing, analizzare il comportamento e le preferenze di utenti sui social serve sicuramente a far migliorare l'offerta. Nascono società che diventano delle vere e proprie "broker di dati" (come ad esempio Cambridge Analytica) raccogliendo informazioni di ogni genere sulle abitudini e i consumi delle persone. La crescita dei social media è stata uno dei grandi fenomeni del 21 ° secolo e la sua evoluzione ha influenzato praticamente ogni aspetto della nostra vita. Dato che è fortemente costruito attorno ai concetti di immagine e apparenza, l'industria della moda è stata senza dubbio uno dei grandi beneficiari.

1.2 Obiettivi

L'obiettivo della tesi è quello di introdurre innanzitutto alla comprensione di come i Big Data possano influenzare mentalità e marketing e, in modo più specifico, all'utilizzo di questi per poter prevedere tendenze del settore fashion e della moda.

In particolare verrà utilizzato un algoritmo di scraping che andrà a "raschiare" dati dal web, nel dettaglio un elenco di immagini da instagram relative a dei post cercati basandosi su specifici tag inerenti al mondo della moda. L'automazione poi successiva di questo sistema di estrazione fa si che questi "data" vengano utilizzati per metodi di deep learning.

1.3 Struttura della tesi

Per quanto riguarda la struttura della tesi, nel secondo capitolo troviamo un'introduzione a ciò che si intende per scraping di dati e come questi vengono modellati ed utilizzati (Data Science); successivamente viene introdotto il concetto di Machine Learning e gli utilizzi che ne vengono fatti.

Nel capitolo tre viene descritto e visualizzato il codice nel dettaglio mostrando anche le varie librerie utilizzate. Nel capitolo quattro vengono elencati svariate potenzialità della libreria utilizzata maggiormente (Selenium) ed esempi di utilizzi di questa. Infine nel capitolo cinque vengono tratte le conclusioni ed eventuali sviluppi futuri che ci possono essere nell'ambito del data science e del machine learning.

2. Stato dell’arte

2.1 Introduzione

L'analisi dei dati sta influenzando le nostre vite. È molto più facile per le aziende analizzare i comportamenti dei propri clienti e le richieste del mercato con dati preziosi in mano. E lo scraping web è il modo migliore per ottenere dati web. Per scraping dei dati, si intende una tecnica in cui un programma informatico estrae dei dati dall'output generato da un altro programma. Lo scraping si esplicita comunemente nel web scraping, che è il processo nel quale una applicazione estrae informazioni di valore da un sito web.

Durante lo scraping vengono raccolti molti tipi diversi di informazioni. Può trattarsi, ad esempio, di informazioni di contatto, come indirizzi di posta elettronica o numeri di telefono, ma anche di singoli termini di ricerca o URL. Questi vengono quindi raccolti in database o tabelle locali.

In genere le aziende non vogliono che i propri contenuti vengano scaricati e riutilizzati per scopi non autorizzati. Di conseguenza, tendono a non esporre tutti i propri dati tramite un'API o altre risorse facilmente accessibili. Chi effettua lo scraping, dall'altra parte, è interessato a ottenere i dati dei siti web indipendentemente da eventuali tentativi di limitazione dell'accesso.

Il processo del web scraping è abbastanza semplice, anche se la sua implementazione può essere complicata.

Gli scraper possono essere progettati per vari scopi, come ad esempio:

1. **Scraping dei contenuti** - i contenuti possono essere estratti dal sito con l'obiettivo di replicare il vantaggio specifico di un dato prodotto o servizio che si basa sul contenuto. Ad esempio, un prodotto come Yelp si basa sulle recensioni; un concorrente potrebbe voler "grattare" le recensioni e riprodurle sul proprio sito, fingendo che il materiale sia originale.

2. **Scraping dei prezzi** - con lo scraping dei dati sui prezzi, le aziende sono in grado di raccogliere e aggregare informazioni sui propri concorrenti, che possono essere utilizzate per costruire una posizione di vantaggio specifica.

3. **Scraping di contatti** - numerosi siti web contengono indirizzi e-mail e numeri di telefono non crittografati. Tramite lo scraping di alcune parti del sito, come ad esempio gli elenchi online dei dipendenti, uno scraper è in grado di raccoglierne le informazioni di contatto, da usare per invio massivo di e-mail, telefonate automatiche, o tentativi di ingegneria sociale dannosi. Questo è uno dei metodi principali con cui spammer e scammer trovano nuovi bersagli.

Esistono diverse tecniche di scraping, ma generalmente si distingue tra scraping manuale e automatico.

Per **scraping manuale** s'intende il processo manuale di copia e incolla di informazioni e dati. Questo può essere paragonato all'attività di ritaglio e raccolta di articoli di giornale. Lo scraping manuale viene eseguito solo se si desidera trovare e memorizzare informazioni singole. È un processo molto impegnativo che viene utilizzato raramente per grandi quantità di dati.

Lo **scraping automatico** invece utilizza invece un software o un algoritmo che ricerca più pagine web per estrarre informazioni. Per questo esistono software specifici, a seconda del tipo di sito web e di contenuti ricercati. Nel caso dello scraping automatico, si distinguono diverse tecniche:

Parser: un parser viene utilizzato per convertire il testo in una struttura nuova. Ad esempio, nell'analisi HTML, il software legge un documento HTML e memorizza le informazioni. Il parsing DOM utilizza la visualizzazione lato client del contenuto nel browser per estrarre i dati. Il Document Object Model

- **Bot:** un bot è un software informatico dedicato a compiti specifici che vengono automatizzati. Il web harvesting utilizza i bot per navigare automaticamente nei siti web e raccogliere dati
- **Text:** se si ha familiarità con la riga di comando, è possibile utilizzare i comandi Unix/grep per cercare termini specifici. Questo è un modo molto semplice per estrarre i dati, ma richiede più lavoro rispetto all'utilizzo di un software.

Lo scraping non sempre è legale e gli scraper (coloro che praticano lo scraping) devono innanzitutto rispettare i diritti d'autore di un sito web. Il web scraping può avere conseguenze piuttosto negative per alcuni negozi e fornitori web, ad esempio, se questo influisce sul posizionamento del sito nei motori di ricerca tramite aggregatori. Non è raro quindi per un'azienda querelare un portale di confronto al fine di prevenire il web scraping.

Il web scraping non è illegale di per sé, si trova in un'area grigia poiché nella maggior parte dei paesi non ci sono leggi chiare a tal riguardo. Questo è uno dei motivi per cui servizi come Instagram hanno al riguardo dei termini di servizio (Terms of use) e se violati portano alla disabilitazione dell'account.

Perché creare uno scraper di instagram? La prima cosa che mi viene in mente sono gli '#' hashtag.
Per molti progetti di machine learning (computer vision o natural language processing), sono necessari dati con una label ed immagini ad alta risoluzione. Poter scaricare immagini con un particolare hashtag sarebbe un risparmio di tempo perché eliminerebbe o ridurebbe il processo di labelling per la creazione del training dataset.

Altro possibile utilizzo di analizzare le immagini è la possibilità di trasformare un contenuto visivo in uno testuale rendendolo accessibile ad uno screen-reader per persone cieche o ipovedenti.

Lo scraping del contenuto testuale, come ad esempio i commenti, può aiutarci ad identificare i migliori commentatori del nostro profilo, la frequenza con cui commentano, le emozioni che esprimono i nostri followers.

Gli scrapers potrebbero avere un accesso più semplice alle informazioni degli account di quanto si possa pensare.

Una massiccia fuga di dati espose più di 300 milioni di account diversi da piattaforme di social media. L'esfiltrazione includeva 192 milioni di record estratti da due diverse raccolte di **Instagram**, insieme a 42 milioni di record estratti da **TikTok** e altri 4 milioni di record estratti da **YouTube**.
I record comprendono nomi utente, foto del profilo, e-mail, numeri di telefono, età e sesso insieme a specifiche sui follower e altri dati per ciascun account. La fuga di notizie ha coinvolto una serie di tre condivisioni di dati aperti dell'azienda Social Data; poche ore dopo la notifica, le condivisioni sono state adeguatamente protette.

Tralasciando i linguaggi e la programmazione, ci sono tantissimi strumenti con cui è possibile estrarre dati dal web tra cui:

- **Octoparse** è uno strumento di scraping potente ed efficace che permette di estrarre diverse tipologie di dati da sorgenti online.

Grazie ad un'interfaccia semplice e visuale è possibile configurare il tool in pochi passi ed impostare l'architettura di estrazione senza dover scrivere una singola riga di codice. **Pro**: molto semplice da utilizzare ma anche potente. Nella versione free permette di estrarre fino . a 10.000 record. **Contro**: purtroppo non offre una versione web ma è necessario scaricare il software stand alone che è compatibile solo con sistemi operativi Windows.

- **Parsehub** è un *software desktop* disponibile per Windows, Mac e Linux dotato di caratteristiche molto avanzate tra cui la possibilità di sfruttare diversi IP (per evitare blocchi da parte del server), l'integrazione con sistemi di archiviazione (come dropbox) e la scansione di siti realizzati con tecnologie come Javascript e Ajax (difficili da scansionare da altri strumenti). Nella versione gratuita Parsehub permette la gestione di 5 progetti e lo scraping di 200 pagine in 40 minuti. **Pro**: strumento con funzioni molto avanzate. **Contro**: è appunto un software desktop e non ha una versione web.

- **Data Miner** è un tool di scraping che si integra con Google Chrome. Tramite l'estensione si possono creare delle ricette di scraping selezionando in maniera visuale i dati da estrarre nella singola pagina. Una volta creata la ricetta si visita il sito e si lancia lo strumento che procede all'estrazione e poi al download delle risorse. Nella versione free lo strumento permette di estrarre fino a 500 pagine al mese. **Pro**: lo strumento è molto semplice da utilizzare e premette l'estrazione di dati in pagine non visibili attraverso un sistema di navigazione in background. **Contro**: il limite di 500 pagine/mese nella versione free può non essere sufficiente per alcuni progetti.

- **Web Scraper** è un'estensione di Google Chrome che si integra con la Console per Sviluppatori. Una volta lanciata, l'estensione permette di creare una sitemap del sito che si vuole "scrapare" selezionando i vari elementi e fornendo un'anteprima del risultato. Al termine della creazione della sitemap basta lanciare l'estrazione e lo strumento ci fornisce una tabella con i dati scaricati esportabile in csv. **Pro**: completamente gratuito e semplice da usare. **Contro**: il sistema è molto basico e non permette estrazioni avanzate.

- **Google Spreadsheets** è il tool di Google dedicato ai fogli di calcolo (la versione Googliana di Excel); lo strumento non nasce come sistema di scraping ma grazie alla funzione *IMPORTXML* permette l'importazione di vari tipi di dati strutturati, tra cui XML, HTML, CSV, TSV e feed XML RSS e ATOM. Nel file spreadsheet va inserito l'url della pagina che si vuole analizzare e le query xpath che vanno ad identificare gli elementi da scansionare. Una volta eseguita la funzione importa nel file Google i dati della pagina che stiamo scansionando. **Pro**: permette la combinazione dei dati importati a qualsiasi altra informazione grazie alle funzioni native dei fogli di calcolo. **Contro**: l'elaborazione dei dati importati ha un limite che non è ben chiaro ma che comunque può creare dei disagi nel caso si debbano importare grandi volumi di dati.

- **ScrapingBee** è una Web Scraper API che mette al servizio dell'utente un headless browser in grado di renderizzare una pagina web (vedendola come la vedrebbe un utente) ed estrarre le informazioni utili per lo scraping. Una volta renderizzata la sorgente ScrapingBee permette di utilizzare librerie Js come React, Angulars e Vue.js per creare degli script di estrazione dei dati. Per evitare blocchi da parte dei siti oggetto dello scraping ScrapingBee offre un servizio di proxies a rotazione che permette agli script di essere eseguiti in modo massivo senza su grandi quantità di dati.

- **ScraperApi** è un servizio pensato per chi fa attività di Scraping in modo massivo, infatti offre una API che permette di gestire attività di proxy rotation, risoluzione di CAPTCHAs, impostazione di headless browsers, in pratica tutto ciò che serve per evitare di essere bloccati durante l'attività di scraping. ScraperApi mette a disposizione dei propri clienti oltre 20 milioni di IP in 12 differenti Paesi offrendo una larghezza di banda illimitata. **Pro**: per i meno esperti di programmazione ScraperBee mette a disposizione degli utenti uno Store API dove scaricare script preconfigurati per eseguire specifiche azioni su tantissimi siti (come Google, Instagram, Booking, etc..). **Contro**: confrontato con altri servizi a parità di piani offre caratteristiche leggermente minori.

Per quanto riguarda invece lo scraping più complesso, andiamo a vedere alcuni dei principali scraper di instagram:

- **Instaloader** è uno script creato principalmente per scaricare immagini o video insieme alle loro didascalie e altri metadati da instagram; lo scraper scarica commenti, geotag e didascalie di ogni post, rileva automaticamente le modifiche al nome del profilo e rinomina la directory di destinazione di conseguenza, consente la personalizzazione a grana fine dei filtri e dove memorizzare i media scaricati, riprende automaticamente le iterazioni di download precedentemente interrotte.

- **Instalooter** è un altro downloader di immagini e video di Instagram senza API; InstaLooter è un programma in grado di scaricare qualsiasi immagine o video associato da un profilo Instagram, senza alcun accesso API. Può essere visto come una reimplementazione dell'ormai deprecato InstaRaider <https://github.com/akurtovic/InstaRaider>_ sviluppato da @akurtovic <https://github.com/akurtovic>_.

- **Instacrawler** è uno script attraverso il quale vengono ottenuti dei dati di post/profilo/hashtag di Instagram senza utilizzare l'API di instagram tramite lo script "**crawler.py**" oppure utilizzare "**liker.py**" per mettere mi piace ai post automaticamente.

La **data science** è una branca del sapere che si fonda su conoscenze relative all'integrazione dei dati, allo sviluppo di algoritmi e alle capacità tecnologiche: di fatto si concentra sulla risoluzione analitica di problemi complessi.

Il cuore della data science, ovviamente, sono i dati e il data scientist è colui che è in grado di utilizzare i dati in modo creativo per generare valore. Un professionista della data scientist coniuga la formazione matematica con competenze informatiche e intuizioni strategiche.

Il compito principale di un data scientist è quello di esplorare i dati. Sulla base di precise domande – tipicamente richieste del business e relative, ad esempio, all'andamento della produzione o delle vendite o alla riorganizzazione delle risorse – il data scientist diventa un vero e proprio investigatore e mette in campo tutta la sua conoscenza in ambito analitico. Tramite l'utilizzo di algoritmi di machine learning riesce ad esaminare e prevedere scientificamente correlazioni tra fenomeni che ad una prima analisi risultano invisibili. Il suo obiettivo è ottenere insights quanto più accurati per fornire al business una panoramica precisa del problema da risolvere.

È possibile che i dati raccolti ne producano altri : la data science nasce per comprendere i dati e analizzarli, ma pure per valorizzarli e far sì che, adeguatamente interrogati e correlati, generino informazioni utili non solo a capire i fenomeni, ma pure ad orientarli.

Amazon, Netflix, Spotify impiegano quotidianamente applicazioni sviluppate dai data scientists che sfruttando intelligenze artificiali sempre più accurate consentono alle macchine di costruire motori di raccomandazione (che ci suggeriscono cosa comprare, cosa guardare, cosa ascoltare in base ai nostri gusti) o di apprendere quali sono le comunicazioni che non desideriamo proprio ricevere.

L'obiettivo della *data science* è quello di sviluppare strategie e modelli per l'analisi dei dati con il fine di ottenere nuove informazioni, ma è pur vero che *data science* e AI sono in un certo senso "complementari". Ad esempio, gli esperti di *data science* si avvalgono spesso dei metodi di deep learning che sono alla base delle reti neurali utilizzate per eseguire operazioni di pulizia dei dati, classificazioni e previsioni. Le applicazioni basate sull'intelligenza artificiale possono poi sfruttare questi dati puliti ed ottimizzati per apprendere come svolgere i propri compiti in modo più efficiente.

L'intelligenza artificiale, infine, permette agli esperti di *data science* di eseguire operazioni di classificazione e di analisi in modo molto più veloce rispetto ad un essere umano e di ottimizzare e velocizzare i processi di estrapolazione delle informazioni dai dati.

Già nel 2001, i cosiddetti "grandi dati" vennero definiti dall'analista Doug Laney come dati caratterizzati da almeno una di queste tre V: volume, velocità o varietà. Si tratta dunque di enormi volumi di dati eterogenei per fonte e formato, spesso da analizzare in tempo reale.

I progetti di Big Data Analytics possono essere classificati secondo quattro tipologie, in base al livello di maturità delle metodologie utilizzate, e di conseguenza alle informazioni che si è in grado di estrarne:

1) Descriptive – cioè le metodologie che descrivono la situazione passata e attuale dei processi aziendali;

2) Predictive – si tratta delle tecniche che effettuano l'analisi dei dati per rispondere a domande relative ad eventi futuri. In questo ambito troviamo tecniche come regressione, forecasting, modelli predittivi. È in questo contesto che può entrare in gioco anche il machine learning;

3) Precriptive – si tratta di modelli che riescono a ipotizzare una serie di scenari futuri. Alcuni esempi di applicazione si hanno nell'ottimizzazione della supply chain e nella manutenzione predittiva;

4) Automated – sono tutti questi strumenti in grado di effettuare autonomamente un'azione sulla base delle analisi di dati effettuate. Esempi sono il dynamic pricing su un sito web o lo smistamento automatico delle pratiche in ambito bancario o assicurativo, con l'obiettivo di identificare le frodi.

5) Gli Advanced Analytics, infine, comprendono le categorie di Predictive, Prescriptive e Automated Analytics. Lo scopo ultimo di queste metodologie è fornire un più ampio supporto ai decisori aziendali, in taluni casi andando ad automatizzare delle specifiche azioni.

I casi d'uso di data science sono fra i più svariati. A titolo di esempio, non esaustivo, si può citarne alcuni, come:

- Nella logistica: il miglioramento dell'efficienza di consegna analizzando i modelli di traffico, le condizioni metereologiche ed altri fattori combinatori;

- Nel marketing retail: la determinazione del tasso di abbandono dei clienti analizzando i dati raccolti dai call center, in modo che si possa agire per tentare di fidelizzarli;

- Nell'Industria 4.0: la predisposizione di piani di manutenzione preventiva allo scopo di ridurre i fermi non programmati, ma anche l'ottimizzazione della supply chain e naturalmente il miglioramento del prodotto;

- In medicina: il miglioramento e la maggiore tempestività delle diagnosi dei pazienti analizzando i dati degli esami clinici e i sintomi segnalati, ma anche l'indagine, attraverso lo studio dei dati provenienti dai social media, di eventuali bisogni in tema di salute pubblica, per finire all'ottimizzazione della ricerca farmaceutica e vaccinale;

- Nella finanza: la rilevazione di frodi, riconoscendo comportamenti sospetti e azioni anomale;

La capacità di sfruttare il potenziale che si nasconde all'interno del mare magnum informativo è un'esigenza sempre più sentita dalle imprese di qualsiasi settore: in base alle stime dell'ateneo meneghino, il 43% dei Chief Information Officer italiani considerava la Business Intelligence, i Big Data e gli analytics tra le principali priorità di investimento per il 2018.

Alla luce dell'enorme mole di dati diversi per tipologia e provenienti da più fonti (datawarehouse aziendali, risorse online, social media, mobile app, dispositivi IoT), il Machine Learning si rivela un anello fondamentale del processo analitico per automatizzare la costruzione, la manutenzione e l'affinamento dei modelli di calcolo. Avere a disposizione algoritmi sempre più attendibili significa aumentare la possibilità di identificare nuove opportunità di profitto, ulteriori margini di efficienza oppure potenziali rischi altrimenti nascosti. Le tecnologie di apprendimento automatico permettono sostanzialmente a un computer di esplorare i dati, dedurre correlazioni e pattern, e, successivamente, delineare dei modelli predittivi. La macchina quindi osserva un campione di dati, ne estrae delle regole e, nel momento in cui esamina altri dati, modifica di conseguenza le proprie conoscenze.

Ciò è possibile grazie all'utilizzo di algoritmi che imparano e si autoregolano sfruttando l'esperienza, attraverso, come detto, l'**analisi iterativa dei dati**. Si tratta di un' insieme di metodi per consentire al software di adattarsi, metodi attraverso i quali si permette alle macchine di apprendere in modo che possano poi svolgere un compito o una attività senza che siano preventivamente programmati.

In altre parole, si tratta di sistemi che servono ad "allenare" l'AI (Artificial Intelligence) in modo che imparando, correggendo gli errori.

Dobbiamo immaginare il ML come un algoritmo neutro che, grazie all'apprendimento autonomo o assistito, imparerà qualsiasi cosa e d'altro canto anche previsioni. Data un'istanza X con determinate caratteristiche, ne calcola le previsioni di tipo Y ; se dovessimo fornire ad un machine learning tutti i dati medico sanitari e referti di centomila individui con caratteristiche simili, il machine learning al termine dell'apprendimento sarà in grado di predire, in maniera più o meno accurata, se un nuovo paziente con caratteristiche analoghe potrà essere soggetto ad una qualche forma di malattia.

Ciò che caratterizza il Machine Learning è quindi il "modello di apprendimento" ed è proprio in base a questi modelli che si può fare una sorta di classificazione degli algoritmi:

– ***con supervisione didattica :*** apprendimento supervisionato mediante esempi di input e di output per far capire all' AI come deve comportarsi

– ***senza supervisione didattica:*** apprendimento basato sulla regressione e analisi dei risultati in questo caso il software capisce come agire e il modello di apprendimento si adatta sulla base di output che permettono di mappano i risultati di determinate azioni e compiti che saranno chiamati a svolgere i software

– ***reinforcement learning*** (apprendimento "meritocratico": l'AI viene premiata quando raggiunge gli obiettivi, i risultati, esegue un'azione, ecc. In questo modo impara quali sono le azioni corrette e quelle errate).

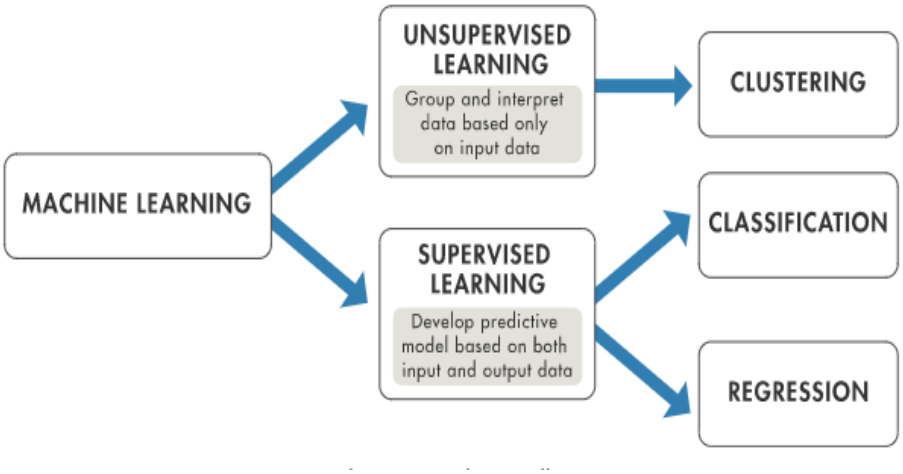

Figura 1.

La complessità si aggiunge quando vado ad immettere più parametri.

Le reti neurali sono usate nel machine learning. Non sono una recente acquisizione. Sono nate diciamo negli anni Cinquanta. Si ispirano al funzionamento del cervello. Gli studiosi di solito criticano questa definizione ma rende bene il senso. In realtà quelli che definiamo neuroni sono numeri il cui valore varia da 0 a 1. Sono raggruppati in layer o livelli. Quelli di primo livello sono incaricati di tradurre in dato come input. Nel caso del riconoscimento di una immagine possiamo pensare al colore di un singolo pixel. Questi informazioni sono connesse ad altre caratteristiche in cui è scomposta l'immagine. Ogni neurone è connesso a un' altro in base a un indice che misura il peso della relazione. L'ultimo layer è formato da neuroni che servono a riconoscere di che animale si tratta e contengono la definizione del singolo animale. Attraverso analisi statistiche sulla base dei pesi delle singole connessioni viene calcolata la probabilità che una data immagina corrisponda alle informazioni contenuti.

Ci sono molte varianti delle reti neurali e negli ultimi anni c'è stata una sorta di boom nella ricerca di queste varianti.

Cos'è un neurone? Tutto quello a cui si dovrebbe pensare è una cosa che contiene un numero in modo specifico un numero compreso tra 0 e 1. Questo numero all'interno del neurone è chiamato attivazione e l'immagine che potresti avere in mente qui è che ogni neurone si illumina quando la sua attivazione è un numero elevato. Passato dal primo strato che contiene x neuroni in base al numero di pixel dell'immagine (i neuroni contengono un valore che rappresenta una scala di grigi del pixel corrispondente) all'ultimo strato che rappresenta quanto il sistema "pensi" che una determinata immagine corrisponda ad una determinata cifra.

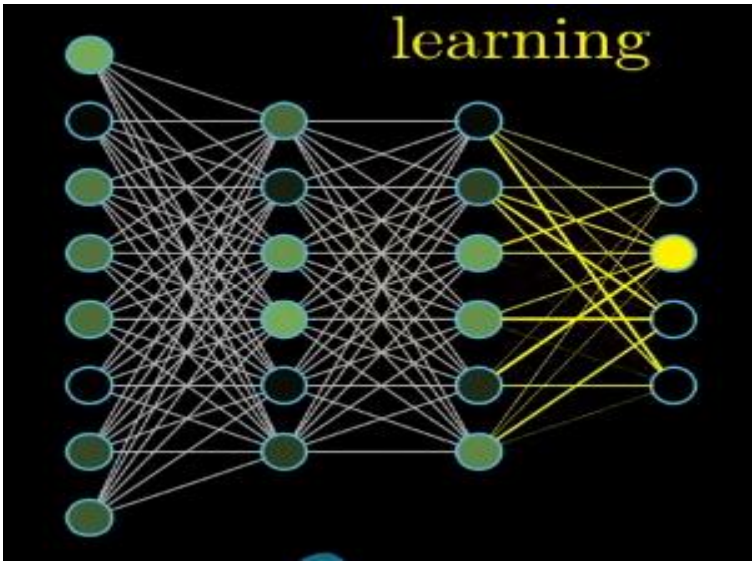

Figura 2.

Ci sono anche dei livelli in mezzo chiamati “hidden layer”; il modo in cui la rete gestisce la attivazioni in un livello determina le attivazioni del livello successivo e chiaramente il cuore della rete come meccanismo di elaborazione delle informazioni si riduce esattamente a come quelle attivazioni di uno specifico livello determinano le attivazioni di quello successivo. È pensato per essere simile al meccanismo con cui alcuni gruppi di neuroni nelle reti neurali biologiche si attivano facendo in modo che si attivino altri neuroni.

Se si alimenta la rete con un’immagine si illuminano tutti i neuroni del livello di input, in base alla luminosità di ogni pixel nell’immagine; quel modello di attivazioni provoca l’attivazione di alcuni schemi specifici nel livello successivo il quale causa qualche altro schema in quello che segue, che alla fine dà qualche altro schema nel livello di output e il neurone più luminoso di questo livello di output è la cifra scelta dalla rete che rappresenta questa immagine.

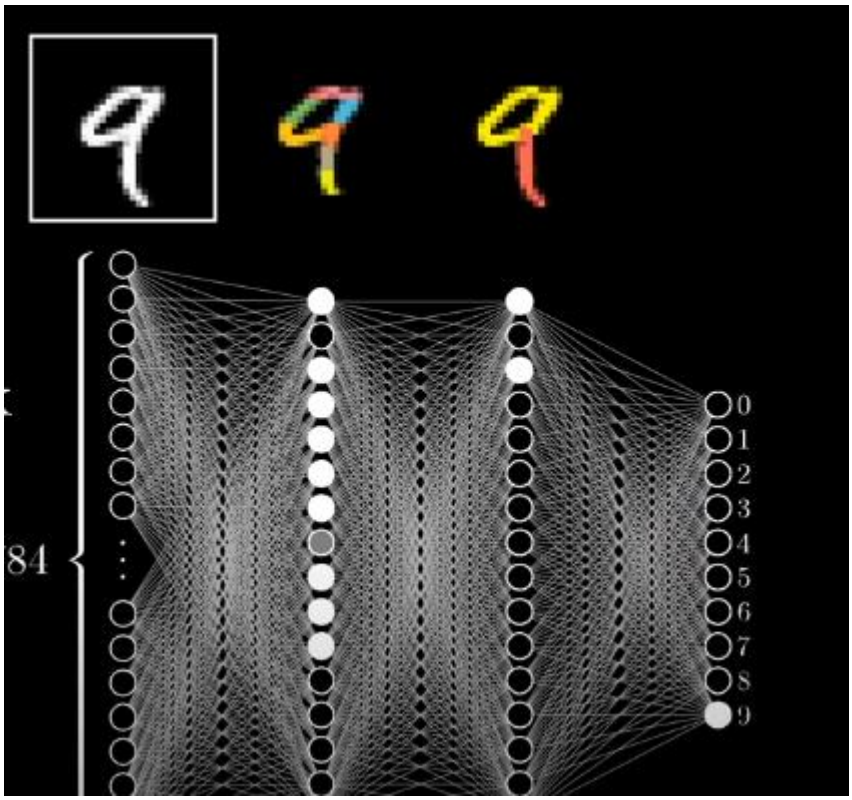

Figura 3

Di base, quando la mente umana riconosce le cifre, mettiamo insieme vari componenti, un **9** ha un cerchio in alto e una linea a destra; un 8 ha anch'esso un cerchio in alto ma è abbinato ad un altro cerchio in basso. L'idea che ci si prospetta è che ogni neurone nel penultimo strato corrisponda a uno di questi sotto-componenti che ogni volta che sottoponi un'immagine con un cerchio in alto come un 9 o un 8 ci siano alcuni specifici neuroni la cui attivazione sia vicina a un sotto-componente.

Passare dal terzo all'ultimo strato è richiesto semplicemente di imparare quale combinazione di subcomponenti corrisponda a quale cifra.

Riconoscere un cerchio è un problema che può essere scomposto in sottoproblemi (tecnica del DIVIDE - ET -IMPERA). Un modo ragionevole per farlo sarebbe innanzitutto riconoscere i vari bordi che lo compongono.

La speranza sarebbe che ogni neurone nel secondo strato della rete corrisponda ai vari piccoli tratti rilevanti dell'immagine ; passata l'immagine (9 ad esempio) si accendono dunque tutti i neuroni associati al cerchio in alto e una lunga linea verticale e alla fine si illuminerà il neurone associato al 9.

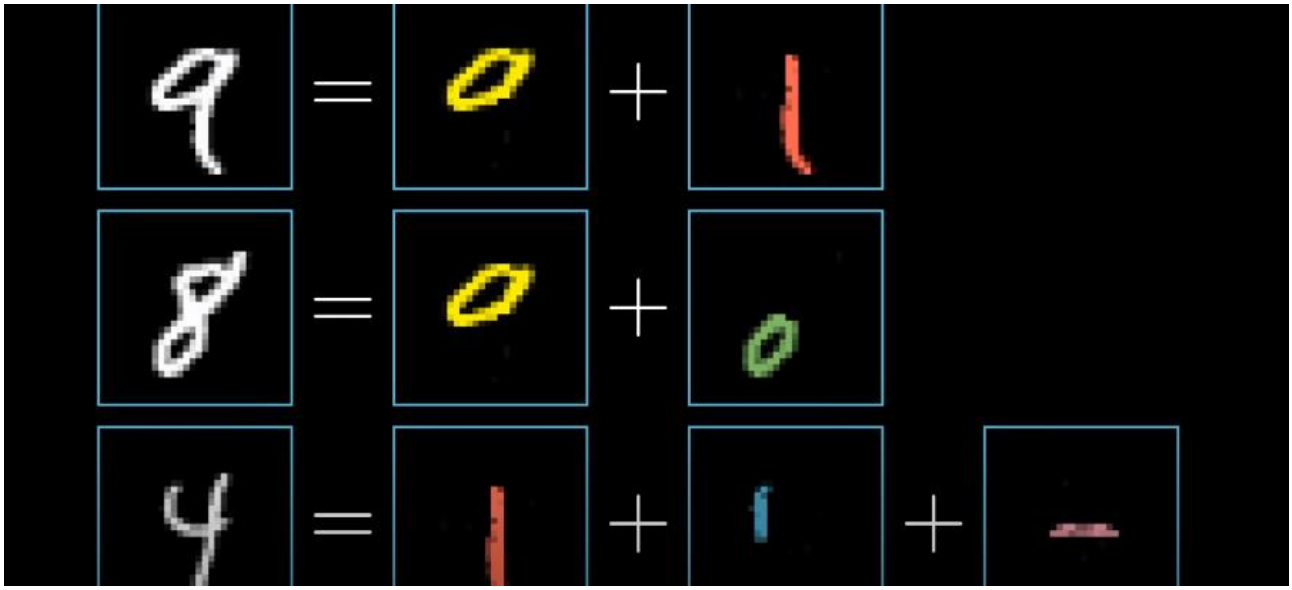

Figura 4

L'obiettivo è avere un meccanismo che possa in teoria combinare i pixel in tratti o i tratti in modelli o modelli in cifre.

Se il Machine Learning può essere definito come il metodo che "allena" l'AI, il Deep Learning è quello che permette di emulare la mente dell'uomo. In questo caso, però, il modello matematico da solo non basta, il Deep Learning necessita di reti neurali artificiali progettate ad hoc (deep artificial neural networks) e di una capacità computazionale molto potente capace di "reggere" differenti strati di calcolo e analisi (che è quello che succede con le connessioni neurali del cervello umano). Può sembrare un livello tecnologico futuristico ma nella realtà questi sono sistemi già in uso nel riconoscimento di pattern, nel riconoscimento vocale o delle immagini e nei sistemi di Nlp – **Natural Language Processing**.

L'obiettivo finale del Natural Language Processing (anche detto elaborazione del linguaggio naturale o NLP) è leggere, decifrare, comprendere e dare un senso ai linguaggi umani in un modo valutabile.

Il machine learning richiede che venga applicata la serie corretta di dati a un processo di apprendimento. Un'organizzazione non deve necessariamente disporre di big data per utilizzare le tecniche di machine learning; tuttavia, i big data possono contribuire a migliorare la precisione dei modelli di machine learning. Con i big data è ora possibile virtualizzare i dati, in modo che possano essere memorizzati nel modo più efficiente ed economicamente conveniente, sia on premise che sul cloud. Inoltre, i miglioramenti della velocità di rete e dell'affidabilità hanno rimosso altre limitazioni fisiche associate alla gestione di enormi quantità di dati a una velocità accettabile.

2.2 Utilizzi pratici

Immaginiamo che i medici vogliano prevedere se un paziente avrà un attacco cardiaco entro un anno. Hanno a disposizione dati di pazienti precedenti, tra cui età, peso, altezza e pressione sanguigna. Sanno se i pazienti precedenti hanno avuto un infarto entro un anno. Il problema, dunque, è combinare i dati esistenti in un modello in grado di prevedere se un nuovo paziente avrà un attacco cardiaco entro un anno.

L'AI ha avuto il pregio di migliorare molti sistemi tecnologici già in uso da persone con disabilità (per esempio i sistemi vocali sono migliorati al punto da permettere una relazione/comunicazione del tutto naturale anche a chi non è in grado di parlare) ma è sul fronte della diagnosi e cura di tumori e malattie rare che si potranno vedere le nuove capacità dell'AI. Già oggi sono disponibili sul mercato sistemi cognitivi in grado di attingere, analizzare e apprendere da un bacino infinito di dati (pubblicazioni scientifiche, ricerca, cartelle cliniche, dati sui farmaci, ecc.) ad una velocità inimmaginabile per l'uomo, accelerando processi di diagnosi spesso molto critici per le malattie rare o suggerendo percorsi di cura ottimali in caso di tumori o malattie particolari. Non solo, gli assistenti virtuali basati su AI iniziando a vedersi con maggiore frequenza nelle sale operatorie, a supporto del personale di accoglienza o di chi offre servizi di primo soccorso.

Un esempio di facile comprensione in ambito sanitario è quello delle immagini, tra cui risonanze magnetiche, raggi X, ecografie e altro ancora. Le immagini mediche sono tra gli strumenti più importanti che i medici usano per diagnosticare anomalie e patologie che vanno dalle semplici fratture ossee fino al cancro. Tuttavia, l'analisi delle immagini mediche può essere un processo difficile e dispendioso in termini di tempo.

L'approccio computazionale può essere applicato non solo alle tecniche di rilevazione di immagini, ma anche per la comprensione dei dati genetici, che non sono interpretabili a occhio nudo, o per lo sviluppo dei farmaci, riducendo tempi e costi delle indagini preliminari grazie alle simulazioni in silico (ovvero al computer con modelli matematici) delle attività della molecola. I dati relativi alle diagnosi e alle terapie sono disponibili sotto forma di cartelle cliniche e tutto ciò che deve essere fatto è inserire i dati abbinati alla diagnosi corretta, così l'algoritmo può imparare.

Parlando di dati, quelli sulla genomica si stanno accumulando in quantità senza precedenti, dando agli scienziati la possibilità di studiare le varianti genetiche, legate a specifiche malattie oppure no. Poiché il costo delle sequenze di DNA diventa sempre più accessibile, lo studio del nostro genoma sembra essere uno dei principali modi di indirizzarci verso la medicina personalizzata, ma l'analisi di questi dati non è per niente scontata. E avere tanti dati senza possibilità di elaborazione equivale a non averli affatto.

Altro esempio può essere il cyberCrime:
la prevenzione delle frodi è una delle applicazioni più mature dove l'Intelligenza Artificiale si concretizza con quelli che tecnicamente vengono chiamati "advanced

analytics", analisi molto sofisticate che correlano dati, eventi, comportamenti ed abitudini per capire in anticipo eventuali attività fraudolente (come la clonazione di una carta di credito o l'esecuzione di una transazione non autorizzata); questi sistemi possono in realtà trovare applicazione anche all'interno di altri contesti aziendali, per esempio per la mitigazione dei rischi, la protezione delle informazioni e dei dati, la lotta al cybercrime.

Per quanto riguarda la pubblica sicurezza, la capacità di analizzare grandissime quantità di dati in tempo reale e di "dedurre" attraverso correlazioni di eventi, abitudini, comportamenti, attitudini, sistemi e dati di geo-localizzazione e monitoraggio degli spostamenti di cose e persone offre un potenziale enorme per il miglioramento dell'efficienza e dell'efficacia della sicurezza pubblica.

Un gruppo di ricercatori dell'Art and Artificial Intelligence Laboratory presso la Rutgers University(NJ,USA) ha voluto verificare se un algoritmo informatico fosse in grado di classificare dipinti per stile, genere e artista con la stessa facilità dell'uomo. Lo studio è iniziato con l'identificazione delle feature visive per classificare lo stile di un dipinto. Gli algoritmi sviluppati hanno classificato gli stili dei dipinti nel database con una precisione del 60%, superando comuni persone non esperte.

I ricercatori hanno ipotizzato che le feature visive utili per la classificazione degli stili (un problema dell'apprendimento con supervisione) potessero essere utilizzate anche per determinare le influenze artistiche (un problema dell'apprendimento senza supervisione). Hanno quindi utilizzato gli algoritmi addestrati con immagini di Google per identificare oggetti specifici e testato gli algoritmi su oltre 1.700 dipinti di 66 artisti diversi vissuti lungo un periodo di 550 anni. L'algoritmo ha identificato con facilità le opere connesse tra loro.

Le somiglianze nella composizione e nel soggetto tra questi due dipinti sono facili da individuare anche per un profano, ma l'algoritmo ha anche prodotto risultati che hanno sorpreso gli storici dell'arte con cui abbiamo lavorato. Ad esempio, il nostro algoritmo ha identificato "Bazille's Studio; 9 rue de la Condamine", dipinto dall'impressionista francese Frederic Bazille nel 1870, come possibile influenza sul "Shuffleton's Barbershop" di Norman Rockwell, completato 80 anni dopo. Sebbene i dipinti possano non sembrare simili a prima vista, un esame più attento rivela somiglianze nella composizione e nei soggetti, compresi i caloriferi in basso a destra di ogni opera, il gruppo di tre uomini al centro e le sedie e gli spazi triangolari nel in basso a sinistra.

Figura 5

Dopo aver identificato in modo affidabile le somiglianze tra coppie di dipinti, eravamo pronti per affrontare la nostra prossima sfida: usare l'apprendimento automatico per rivelare le influenze artistiche. La nostra ipotesi era che le caratteristiche visive utili per la classificazione dello stile (un problema di apprendimento supervisionato) potessero essere utilizzate anche per determinare le influenze (un problema non supervisionato).

Gli storici dell'arte sviluppano teorie sull'influenza artistica basate su come gli artisti hanno lavorato, viaggiato o si sono formati con i contemporanei.

2.3 Utilizzi Software

Secondo il grafico di Francois Puget, del dipartimento di machine learning di IBM, Python è il principale linguaggio di codice per AI e ML.
Nello specifico offre una vasta scelta di librerie open-source ed è uno dei motivi principali per cui python è il linguaggio di programmazione più utilizzato per l'AI.
Una libreria è un modulo o un gruppo di moduli pubblicati da diverse fonti come **PyPy** (repository per il linguaggio di programmazione python) che includono un pezzo di codice pre-scritto che consente agli utenti di raggiungere alcune funzionalità o eseguire azioni diverse.

La libreria **pandas** serve a caricare dei dati da disco e tenerli in RAM del computer, effettuare azioni o modellamenti e poi salvarli. **Keras** è una libreria per le reti neurali e il deep learning. Sono librerie molto veloci perché anche se i comandi sono in python, le routine più interne sono scritte in c o c++. Sono Open Source quindi il codice è visibile a tutti e può essere utilizzato senza pagare nessuno; si può contribuire anche a farle crescere. Sono librerie facilmente reperibili online e molto documentate spiegando quali sono i comandi a disposizione, quali sono le opzioni di questi comandi. Molto supportate essendoci una grande community intorno a queste librerie è facile trovare il modo di documentarsi come ad esempio su stackoverflow e vari forum.

Con pandas è possibile caricare file da disco in diversi formati che possono essere CSV, EXCEL, SQL, JSON ed è possibile fare su questi dati operazioni di tutti i tipi spostare le colonne, addizionarle, moltiplicarle, incrociare tabelle su dati mancanti.
Keras è un wrapper di tensor.flow che al suo interno utilizza tensorflow di google che

serve per calcolo distribuito, reti neurali e deep learning; è possibile costruire reti multistrato oppure reti a convoluzione per le immagini.

Il codice delle librerie richiama classi e metodi che normalmente definiscono operazioni specifiche in un'area del dominio.

Ad esempio, ci sono alcune librerie di matematica che possono far sì che lo sviluppatore chiami semplicemente la funzione senza ripetere l'implementazione di come funziona un algoritmo. **NumPy** sta per Numeric Python e rappresenta il pacchetto fondamentale per il calcolo scientifico con Python. NumPy è ovviamente una delle più grandi librerie di calcolo matematico e scientifico per Python.

Una delle funzionalità più importanti di NumPy è la sua interfaccia Array. Questa interfaccia può essere utilizzata per esprimere immagini, onde sonore o altri flussi binari grezzi come matrici di numeri reali con dimensione N. La conoscenza di NumPy è molto importante per l'apprendimento automatico e la scienza dei dati.

Secondo diversi studi **Tensorflow** è una delle librerie più utilizzate da chi si affaccia nel mondo del deep learning. Tale libreria è stata sviluppata da Google; se stai utilizzando le foto di Google o la ricerca vocale di Google, indirettamente stai utilizzando i modelli creati utilizzando Tensorflow.

Tensorflow è solo un framework computazionale per esprimere algoritmi che coinvolgono un gran numero di operazioni di tensori, poiché le reti neurali possono essere espresse come grafici computazionali tramite una serie di operazioni sui tensori. I tensori sono matrici N-dimensionali che rappresentano i nostri dati.

Microsoft Azure, piattaforma di cloud computing di Microsoft, dispone di un'ampia libreria di algoritmi di famiglie di classificazione, sistemi di raccomandazione, clustering, rilevamento di anomalie. Ognuna è progettato per risolvere un tipo diverso di problema di Machine Learning.

Gli utenti possono scegliere tra questi servizi per sviluppare e scalare nuove applicazioni o eseguire applicazioni esistenti nel cloud. La piattaforma Azure mira ad aiutare le aziende a gestire le sfide e raggiungere i propri obiettivi organizzativi. Offre strumenti che supportano tutti i settori, inclusi e-commerce, finanza ed è compatibile con le tecnologie open source. Al programma Microsft Azure per Machine Learning si accede tramite l'utilizzo di diversi linguaggi Java, Python e JS.

Le macchine di apprendimento possono essere sviluppate utilizzando anche un'altra applicazione in cloud, fornita da Google e denominata **Google Prediction** (Google Cloud Prediction API per Machine Learning).

Di fatto, il kit di strumenti per il Machine Learning di Google appartiene alla suite di prodotti Google in cloud, raggiungibili sul Google Cloud Platform. Per il Machine Learning, Google fornisce la piattaforma di sviluppo della macchina di apprendimento in cloud, le API Cloud Jobs (ad oggi disponibili nella loro versione "Alpha"), le API Cloud Natural Language, le API Cloud Speech, le API Cloud Translation e le API Cloud Vision.

Gli strumenti di sviluppo di una macchina di apprendimento con Google sono molto efficaci, il loro utilizzo viene determinato dalle necessità che hanno portato a programmare la macchina di apprendimento e alla tipologia di dati che devono essere analizzati e previsti. Google è molto attiva nel campo del Machine Learning per il suo motore di ricerca e per il miglioramento degli strumenti messi a disposizione dei suoi utenti. Un software in cloud computing per lo sviluppo di una macchina di apprendimento può migliorare l'esperienza di navigazione su internet ed offrire prodotti più sicuri ed intelligenti, ottimizzando i processi e rendendo più consapevole l'interazione uomo macchina.

3 Materiali e metodi

3.1 Packages

Per la realizzazione del mio bot di scraping sono partito dal linguaggio python. Come libreria ho usato maggiormente "Selenium"; il software libero Selenium è un framework per il test automatizzato dei software delle applicazioni web.

Progettato per testare siti e applicazioni web, Selenium WebDriver può anche essere utilizzato quindi con Python per il web scraping. Sebbene Selenium stesso non sia scritto in Python, è possibile accedere a questa funzione del software tramite il linguaggio di programmazione.

A differenza di Scrapy o BeautifulSoup, Selenium non opera sul codice sorgente HTML della pagina. La pagina viene invece caricata in un browser (simulazione di una pagina browser) senza interfaccia utente.

Il browser interpreta il codice sorgente della pagina e crea un Document Object Model (DOM). Questa interfaccia standardizzata consente di testare le interazioni utente e permette, tra le altre cose, di **simulare clic e compilare i moduli automaticamente**. Le conseguenti modifiche del sito si riflettono nel DOM. Il web scraping con Selenium avviene quindi in questo modo:

URL → Richiesta HTTP → HTML → Selenium → DOM

Poiché il DOM viene generato in modo dinamico, Selenium può effettuare anche lo scraping di pagine i cui contenuti sono stati generati tramite JavaScript. L'accesso ai contenuti dinamici è il vantaggio principale di Selenium. Selenium può essere utilizzato anche in combinazione con Scrapy o BeautifulSoup.

In prima battuta, Selenium fornisce il codice sorgente, mentre il secondo strumento (Scrapy/BeautifulSoup) si occupa dell'analisi e della valutazione. In questo caso, il processo diventa:

URL → Richiesta HTTP → HTML → Selenium → DOM→ HTML → Scrapy/BeautifulSoup

3.2 Source code e librerie

Python gode di diverse funzionalità che lo rendono il linguaggio più adatto per il web-scraping. Python è semplice da codificare: non è necessario aggiungere punti e virgola ";" o parentesi graffe "{}" ovunque. Questo lo rende meno disordinato e facile da usare. Usufruisce di una vasta raccolta di librerie che forniscono metodi e servizi per vari scopi. Quindi, è adatto per il web scraping e per l'ulteriore manipolazione dei dati estratti.

La digitazione automatica di python consente di non definire i tipi di dati per le variabili ed è possibile utilizzarle direttamente dove richiesto; ciò consente di risparmiare tempo e velocizza il tuo lavoro. La sintassi di python è facilmente comprensibile principalmente perché leggere un codice python è molto simile alla lettura di un'istruzione in inglese. . È espressivo e facilmente leggibile e il rientro utilizzato in python aiuta anche l'utente a distinguere tra diversi ambiti/blocchi nel codice.

Il web scraping viene utilizzato per risparmiare tempo. In python puoi scrivere piccoli codici per svolgere compiti di grandi dimensioni. Quindi risparmi tempo anche durante la scrittura del codice.

MyScraper.py

```
import os
import time


import wget
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import WebDriverWait


import sys



DRIVEPATH =
"C://Users//Sony//PycharmProjects//igScraper//chromedriver_win32//chromedriver.exe"
URL = https://www.instagram.com/







key_username = sys.argv [1]
key_pssw = sys.argv [2]
tag = sys.argv [3]


name_username = "username"
name_password = "password"
class_cookie = "aOOlW.bIiDR"
class_log = "sqdOP.L3NKy.y3zKF"
class_NonOra = "sqdOP.yWX7d.y3zKF"
class_notifiche = "aOOlW.HoLwm"




def class_search(driver,class_name):
    try:
        obj  = WebDriverWait(driver, 8).until(
            EC.presence_of_element_located((By.CLASS_NAME, class_name))
        )

    finally:
        time.sleep(5)
```

```
    return obj


def name_search(driver,class_name):
    try:
        name = WebDriverWait(driver, 8).until(
            EC.presence_of_element_located((By.NAME, class_name))
        )

    finally:
        time.sleep(5)

    return name



def selector_keys(driver,class_name,key):
    try:
        name = name_search(driver,class_name)
        name.send_keys(key)

    finally:
        time.sleep(5)



def selector_click(driver,class_name):
    try:
      clk =  class_search(driver,class_name)
      clk.click()

    finally:
        time.sleep(5)



def keyGo(driver,class_name,tag_name):
    try:
        search = class_search(driver,class_name)
        search.send_keys(tag_name)
        count=0
        while count <= 2:
            go(search)
```

```
            count = count + 1

    finally:
        time.sleep(8)



def go(obj):
    obj.send_keys(Keys.ENTER)
    time.sleep(8)



def scrape(driver):
    driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")

    posts = driver.find_elements_by_tag_name('img')
    posts = [image.get_attribute('src') for image in posts]
    posts = posts[:-2]

    path = os.getcwd()
    path = os.path.join(path, "C://Users//Sony//PycharmProjects//igScraper//" + tag)
    os.mkdir(path)

    counter = 0
    for post  in posts:
        save_as = os.path.join(path,  str(counter) + '.jpg')
        wget.download(post,save_as)
        counter += 1







def ig_panel(driver):
    class_search = "eyXLr.wUAXj"
    button_class_search = "XTCLo.x3qfX"
    selector_click(driver,class_search)
    keyGo(driver,button_class_search,tag)


def log(driver):

        driver.get(URL)
        selector_click(driver, class_cookie)
```

```
        selector_keys(driver, name_username, key_username)
        selector_keys(driver, name_password, key_pssw)

        selector_click(driver, class_log)
        selector_click(driver, class_NonOra)
        selector_click(driver, class_notifiche)



def main():
    try:
        driver = webdriver.Chrome(DRIVEPATH)
        log(driver)
        ig_panel(driver)
         scrape(driver)
    finally:
            time.sleep(5)
            driver.quit()


main()
```

All'interno del codice come prima cosa si va a trovare la path dove è stato scaricato Selenium WebDriver. Selenium WebDriver è l'ambiente di sviluppo che permette di registrare, modificare e debuggare test per web application. E' implementato come un driver per browser e non necessita di un server per eseguire i comandi. Ciò gli permette di interagire con i browser più diffusi come Firefox, Chrome, Internet Explorer e Microsoft Edge. I Test sono registrati in Selenese, uno speciale linguaggio di scripting ideato appositamente per Selenium.

Viene attivato il driver di selenium associato al browser google chrome, passando al metodo .chrome la path dove si trova l'eseguibile "chromedriver.exe".

Il driver servirà poi alle altre funzioni per poter interagire con la DOM e quindi con la pagina web. Allocato in una variabile il driver, viene lanciata la funzione "log()" la quale inizialmente simula la pagina browser (passando come parametri l'URL della pagina da simulare) ed inizia le interazioni con questa. Successivamente sono richiamate delle funzioni che hanno il compito di eseguire delle azioni sulla pagina: la funzione "selector_click()" viene utilizzata per interagire con la pagina web e i vari pop-up facendo uso dei tag html presenti nella DOM;

alla funzione viene passato il driver di selenium e di volta in volta il tag html da analizzare. In "selector_click()", dopo aver associato ad una variabile il contenuto del tag HTML tramite il nome della classe passato come parametri ad una funzione di ricerca, "class_search()", effettua come operazione un click in corrispondenza dell'oggetto sulla pagina browser.

L'altra funzione che troviamo di seguito è "selector_keys()"; viene sempre utilizzata per interagire con la pagina web facendo uso dei tag HTML, ma in questo caso ciò che va a fare è la compilazione di moduli, popolamenti di credenziali o comunque l'immissione di una chiave di valori (intese nel software come "key").

Ora ci troviamo nella home page di Instagram dell'account della vittima.
A questo punto, "clickato" l'elemento della DOM che rappresenta la barra "cerca", viene lanciata un'altra funzione che popola la barra in questione con delle credenziali (in questo caso vogliamo fare lo scraping di post che contengono un determinato tipo di hashtag) e successivamente viene "premuto" il tasto "INVIO".

Essendo quindi nella sezione "Esplora" di Instagram che propone la visualizzazione dei post contenenti il tag specifico, vengono ispezionati i posts dalla Dom così da poter iniziare lo scraping.

La funzione lanciata "scrape()" parte scorrendo la pagina dall'inizio alla fine raccogliendo quanti più posts possibili; i posts vengono poi salvati in una lista attraverso l'utilizzo dei metodi del WebDriver servendosi anche qui dei vari tag che rappresentano i post sulla pagina browser.

Successivamente viene creata una cartella (dello stesso nome dell'hashtag in questione) nella directory preferita; i post vengono scorsi tramite un ciclo for, scaricati e salvati in formato ".jpg".

3.2 Scraping

Lo scopo del Document Object Model è rendere il più facile possibile ai programmatori l'accesso ai componenti di un progetto web e quindi aggiungere, cancellare o modificare contenuti, attributi e stili. Il DOM funge da collegamento tra linguaggi di scripting come JavaScript e il documento web sottostante, indipendentemente dalla piattaforma e dal linguaggio,

rappresentando la struttura del documento in una **struttura ad albero** dove ogni nodo è un oggetto indipendente e gestibile.

Il Document Object Model è stato sviluppato per il **World Wide Web** e viene utilizzato principalmente in questo ambito. Più precisamente, sono i browser con cui gli utenti accedono alle offerte del web che si avvalgono dell'interfaccia standardizzata: ad esempio, i comuni web client utilizzano DOM o interfacce basate su DOM per visualizzare pagine **HTML** o **XML** da consultare. In questo processo, i singoli componenti sono raggruppati in nodi e organizzati in un singolo albero DOM. Allo stesso tempo, il rispettivo browser carica questa versione mostrata del documento web nella **memoria locale** per analizzarla o elaborarla e infine presentare la pagina nella forma voluta dallo sviluppatore.

I markup HTML definiscono le relazioni tra i vari tag che contengono. Ad esempio, gli elementi taggati di un documento web sono collegati tra loro a seconda del ruolo che ricoprono nel progetto. Inoltre, alcuni tag possono essere contenuti in altri tag. Per riflettere adeguatamente queste **gerarchie** nel Document Object Model, l'interfaccia utilizza una struttura ad albero, che permette di disporre nel modo corrispondente gli oggetti visualizzati.

La **struttura di un albero DOM** dipende quindi sempre dal documento HTML o XML sottostante. Nel primo caso, tuttavia, si può definire la seguente **gerarchia di base** valida per più progetti:

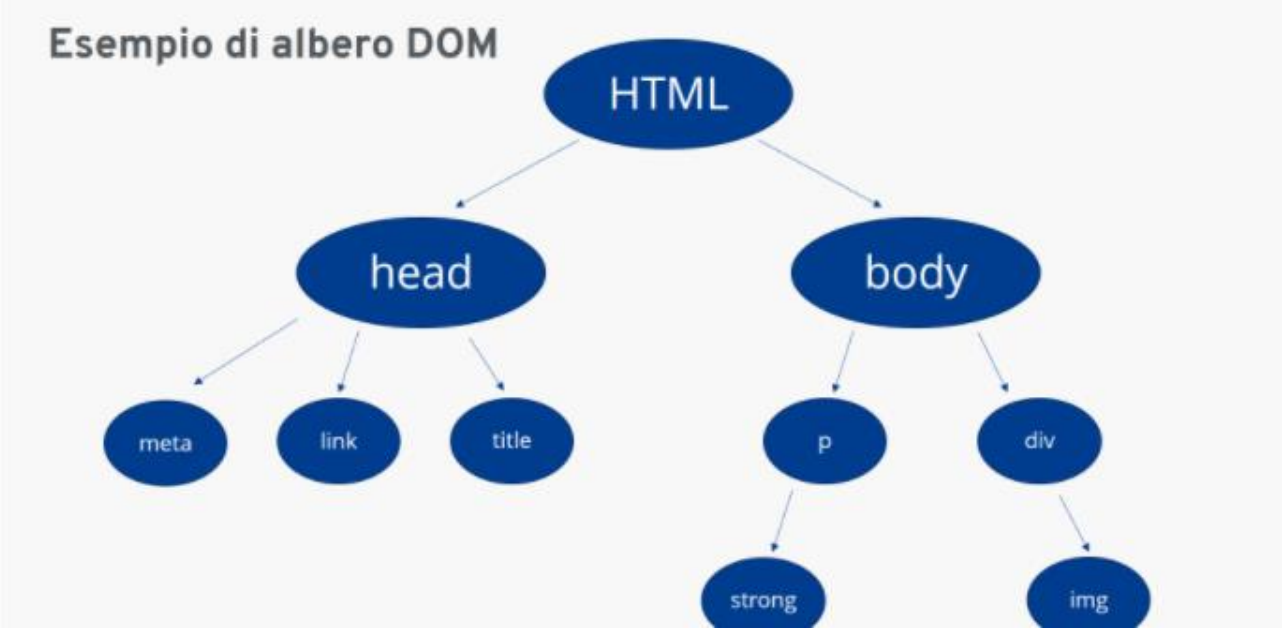

Figura 6

Questo ci aiuta a trovare e indirizzare **tag** , **id** o **classi** cruciali all'interno del documento ed estrarre lo stesso. Per riassumere, DOM è uno standard che ci consente di:

- **ottenere**
- **modificare**
- **aggiungi**
- **Elimina**

elementi HTML.

Per ispezionare i singoli elementi all'interno di una pagina web, possiamo utilizzare il DOM inspector (o le sue varianti) che viene fornito con ogni browser.

Il modo più semplice per accedere al codice sorgente di qualsiasi pagina Web è tramite la console facendo clic su **F12**

In alternativa, possiamo fare clic con il pulsante destro del mouse su un elemento specifico nella pagina Web e selezionare **ispeziona** o **ispeziona elemento** dal menu a discesa. Ciò è particolarmente utile nei casi in cui vogliamo indirizzare un dato specifico presente all'interno di un elemento HTML. Ciò aiuta a evidenziare diversi attributi, proprietà e stili all'interno dell'HTML.

La localizzazione dei dati su un sito web è uno dei principali casi d'uso di Selenium, sia per una suite di test (assicurandosi che un elemento specifico sia presente/assente nella pagina) o per estrarre i dati e salvarli per ulteriori analisi (web scraping).
Ci sono molti metodi disponibili nell'API Selenium per selezionare gli elementi nella pagina. Puoi usare:

- Nome dell'etichetta
- Nome della classe
- ID
- XPath
- Selettori CSS

Il selenio è anche un ottimo strumento per automatizzare quasi tutto sul web. Se esegui attività ripetitive come compilare moduli o controllare le informazioni dietro un modulo di accesso in cui il sito Web non ha un'API, è forse * una buona idea automatizzarlo con Selenium.

3.4 Interazioni ed utilizzi

Selenium WebDriver è lo strumento che simula il comportamento di un utente reale all'interno di un browser : per questo banalmente Selenium non viene utilizzato per lo scraping ma per la navigazione appunto. Ed è per questo che può essere utilizzato per automatizzare qualsiasi tipo di operazione su una pagina browser; per esempio, nel nostro caso, invece di andare ad analizzare i post ed effettuare di seguito lo scraping si va ad analizzare i messaggi di posta privati. A quel punto si vanno a leggere le chat e si inizia lo scraping dei data relativi a queste, come ad esempio testo dei messaggi, profilo di chi li invia, profilo del destinatario…
Volendo invece inviare un messaggio utilizzando Selenium, è possibile usufruire dei metodi della suddetta libreria per l'immissione di un testo o una chiave di valori interagendo con l'oggetto della Dom relativo all'immissione dei caratteri di input e, successivamente, il "click" su invia per inviare appunto il messaggio.

Con Selenium è quindi possibile non solo creare bot per lo scraping di vari data da vari social e pagine web, ma anche automatizzare meccanismi basilari o più complessi di autenticazione, gestione, controllo di ogni pagina richiesta a partire dal suo URL. Un esempio potrebbe essere il controllo dei prezzi altalenanti di vari articoli specifici che si tengono d'occhio, oppure quando è necessario fare accessi continui ad un sito o pagina browser.
È possibile anche aprire più finestre contemporaneamente e gestire entrambe dal proprio software. Principalmente però l'utilità di Selenium attinge alla capacità di simulare pagine web, testarle e quindi ispezionarle, aiutando così sviluppatori che magari devono testare i propri siti e pagine web da un punto di vista della funzionalità e della sicurezza.

4. Conclusioni e sviluppi futuri

L'incontro tra intelligenza artificiale e industria della moda è scritto nel destino. La straordinaria crescita di interesse per il tema, del resto, è sotto gli occhi di tutti. Sappiamo che il mercato dell'intelligenza artificiale raggiungerà un valore di 90 miliardi di dollari nel 2025. Nell'ambito del fashion, le "top 10 apparel companies" (secondo la classificazione di Forbes) hanno fatto da apripista, sposando la logica del supporto algoritmico in vari campi. Dal design alla produzione, dal marketing al customer care. Il bisogno di AI è in gran parte legato alla necessità di offrire al cliente un'esperienza gratificante e personalizzata a un costo sostenibile.

Grazie alle tecniche di machine learning, le aziende che operano nel settore della moda possono individuare i pattern nei dati e costruire modelli in grado di prevedere risultati futuri. Questo contribuisce a creare una supply chain più flessibile e più veloce e gestire l'inventario in modo automatizzato e intelligente. Gli strumenti AI-powered possono aiutare i rivenditori a ridurre gli errori di proiezione fino al 50%, alleggerendo, al contempo, le scorte dal 20% al 50%.

H&M, leader mondiale del fast fashion, ha in programma l'impiego di tecnologie di machine learning per personalizzare la sua offerta in relazione all'area geografica in cui è presente. L'analisi predittiva si basa sui dati in-house – quelli relativi alle vendite, ai resi, al profilo del cliente – e sui Big Data. Adottando questo approccio, la multinazionale svedese riesce a comprendere meglio le tendenze locali e la richiesta del mercato, e quindi rispondere in modo più pertinente.

Amazon ha sviluppato un algoritmo che è in grado di disegnare i capi di abbigliamento attraverso l'image recognition (riconoscimento delle immagini). L'addestramento del software si basa sulla raccolta di una quantità enorme di raw data – immagini, testi o suoni – per poi fornire un modello che è in grado di generare dati simili. L'analisi pone accento sul pop up di prodotto che va di moda nei post sui social media, vista l'importanza di questi ultimi al giorno d'oggi.

Fashion Flair è la prima collezione di moda creata dall'AI di Huawei in collaborazione con Anna Yang, creative director di Annakiki, brand che ha già sfilato per 5 anni nelle Fashion Week milanesi. Il progetto – spiega la big tech cinese – vuole dimostrare la possibilità di dare vita a una collezione geniale facendo partecipare l'algoritmo.

«Fornendo i parametri di base per la realizzazione di un abito, come colore, lunghezza, volume e texture, l'Intelligenza Artificiale è ora in grado di fornire uno spunto creativo ai designer da cui partire per poi realizzare le loro creazioni» spiega Isabella Lazzini, marketing e retail director di Huawei Cbg Italia.

L'omonima app, Fashion Flair, è stata sviluppata insieme a un'equipe di programmatori italiani.

Spesso questo settore viene considerato da molti troppo creativo e soggettivo per dei sistemi informatici ma i confini dell'A.I. sono riusciti ad abbattere queste credenze.

Nasce il sistema Fashion ++, un sistema di Intelligenza artificiale che mira a suggerire piccole alterazioni agli abiti al fine di renderli più alla moda.

Fashion++ utilizza una deep neural network basata sulle immagini, in grado di riconoscere i capi indossati e suggerire cosa togliere, aggiungere o cambiare sul tuo outfit. È anche in grado di consigliare piccoli accorgimenti, come arrotolare le maniche, in grado di dare una nuova forma al tuo look.

Fashion++ si concentra principalmente su modifiche minimali, suggerendo accorgimenti che siano più realistici e pratici di proporre o di cambiare interamente l'outfit. Il sistema utilizza un classificatore in grado di identificare quanto sia fashion il look preso in esame, anche grazie all'allenamento su migliaia di immagini pubbliche che sono state giudicate alla moda. Queste servono come base per la creazione di possibili outfit adeguati, mentre gli esempi non conformi vengono modificati scambiando accessori e vestiti con le controparti meno simili a loro. Una volta che il classificatore è allenato, il sistema aggiorna gradualmente il nostro outfit per renderlo più trendy. Una rete neurale generatrice di immagini disegna il nuovo look aggiornato, utilizzando vari algoritmi e soluzioni per generare la silhouette e, successivamente, riempirla con colori e pattern. I parametri utilizzati ed imparati da questo generatore sono poi utilizzati anche per comprendere quali capi possano essere i più adatti, tra quelli disponibili.
Alcuni esperimenti hanno dimostrato che le raccomandazioni generate dal sistema sono in grado di creare immagini molto simili alla realtà e che risultano anche facili da riprodurre con i propri indumenti.

Non sorprende che le tecniche di ML diano una mano anche nella creazione di sistemi che commerciano nei mercati delle criptovalute e del trading. Thomas E. Koker e Dimitrios Koutmos hanno scritto un documento di ricerca che ha presentato l'uso dell'apprendimento per rinforzo diretto per creare un modello per il trading attivo basato sulla criptovaluta.

Reinforcement Learning (RL) è un sottodominio del Machine Learning comune all'interno dei giochi e dei programmi di simulazione. RL opera attraverso programmi di formazione (agenti) sullo sviluppo di una strategia ottimizzata (politica) per ottenere ricompense all'interno di un ambiente interattivo.

L'approccio presentato in questa particolare ricerca fa leva sul Direct Reinforcement Learning (DRL). Non c'è un feedback immediato sulle prestazioni per l'agente in RL tradizionale, ma all'interno di DRL, le prestazioni delle finestre precedenti vengono utilizzate come feedback per l'agente. Utilizzando DRL, i ricercatori sono stati in grado di eliminare la creazione di un modello di previsione dei prezzi e hanno creato un sistema che si adatta in base a un intervallo di tempo specificato (giornaliero).

l'applicabilità delle tecniche ML per ottimizzare gli sforzi di mining e prevenire il dirottamento delle risorse minerarie.

Taotao Wang , Soung Chang Liew e Shengli Zhang sono autori di un documento di ricerca, pubblicato nel gennaio 2021, che presenta l'applicazione dell'apprendimento di rinforzo (RL) per l'ottimizzazione della strategia di mining blockchain per criptovalute come Bitcoin.

Il loro sforzo di ricerca ha dimostrato che senza un modello iniziale della blockchain e dei parametri corrispondenti (risorse di calcolo del minatore, commissioni di transazione ecc.) è stato possibile sfruttare le tecniche RL per estrapolare dinamicamente strategie di mining più performanti rispetto ad altre strategie (estrazione tradizionale onesta e l' estrazione egoistica).

I tradizionali algoritmi di apprendimento per rinforzo escogitano metodi con cui gli agenti possono massimizzare l'ottenimento di ricompense in un ambiente. Ma la rete blockchain è un ambiente dinamico in cui non è possibile creare facilmente un modello rappresentativo. Gli autori dell'articolo hanno ideato un algoritmo RL multidimensionale che utilizza Q Learning (algoritmo model-free) per ottimizzare il mining di criptovalute.

Gli autori hanno dimostrato che attraverso tecniche di apprendimento automatico è possibile risolvere lo sviluppo di strategie di mining performanti. Non è un segreto che il mining di bitcoin e criptovalute sia un settore in forte espansione. Secondo quanto riferito, diverse società minerarie come Argo Blockchain, Riot Blockchain e Hive Blockchain hanno estratto diversi milioni di Bitcoin. Queste stesse aziende potrebbero avere ingegneri ML tra i loro team tecnici.

Il Machine Learning ha un posto nel mondo delle blockchain e delle criptovalute. L'applicabilità delle tecniche ML va oltre la previsione o il trading dei prezzi delle criptovalute.

Alcuni degli svantaggi che si riscontrano comunemente anche nel campo del processo di apprendimento automatico.

Quei fattori che hanno un impatto in ML sono i seguenti:

Acquisizione dei dati

Nel processo di apprendimento automatico, una grande quantità di dati viene utilizzata nel processo di formazione e apprendimento.

Quindi questo uso dei dati dovrebbe essere di buona qualità, imparziale. Durante il processo di apprendimento automatico con l'aiuto dei servizi di sviluppo software , ci sono anche momenti in cui dobbiamo aspettare. In quel periodo di tempo vengono generati nuovi dati che possono essere utilizzati per ulteriori processi.

Tempo e risorse

Durante la procedura di machine learning elaborano gli algoritmi che aiutano a gestire tutte le funzioni per gestire i dati e l'utilizzo di determinati dati nel processo di rettifica se eventuali errori tutto questo richiede tempo. E anche risorse affidabili e affidabili per il funzionamento di questo sistema.

Interpretazione

Quando gli algoritmi aiutano in tutti questi processi e danno un output risultante. Questo dato output deve essere controllato per eventuali errori e l'operazione di correzione deve essere seguita per ottenere la precisione desiderata. E durante la selezione di questo algoritmo, dobbiamo selezionare l'algoritmo di cui hai bisogno per lo scopo.

Elevata suscettibilità agli errori

Nel processo di apprendimento automatico, viene utilizzata l'elevata quantità di dati e, d'altra parte, vengono utilizzati e testati molti algoritmi. Quindi c'è un enorme cambiamento per sperimentare molti errori. Perché mentre stai addestrando il tuo set di dati su quel particolare, molti algoritmi vengono utilizzati se c'è qualche errore nell'algoritmo, allora può portare l'utente a diversi annunci pubblicitari irrilevanti.

Questi errori sono un problema comune che si verifica molte volte. Perché quando si verificano questi errori, non è facile scoprire la fonte principale per cui è stato creato il problema e scoprire quel particolare problema e risolverlo richiede più tempo.

Una macchina non ha l'esperienza di un programmatore e non riesce a contestualizzare il requisito. L'ultimo test effettuato ha però mostrato che diminuendo il livello di difficoltà del problema da risolvere si possono ottenere dei risultati discreti. L'idea sarebbe quindi di raggiungere i nostri obiettivi con una metodologia top-down, riducendo ciò che si vuol ottenere nelle componenti più elementari, utilizzando i risultati stessi come input e determinando infine le stime cercate.

Ringraziamenti

Vorrei ringraziare prima di tutto la mia famiglia, che mi ha aiutato e sostenuto, tramite sforzi e sacrifici, non solo in questi anni ma durante tutta la vita. Grazie per essermi stati vicini quando non lo meritavo e per avermi permesso di intraprendere questo percorso.

Ringrazio gli amici con cui ormai ho passato una vita insieme, qualcuno mi accompagna dalle medie o dai primi anni di superiori ma tutti mi conoscete benissimo. Non sono bravo con le smancerie, né coi contatti fisici, ma sapete che vi voglio un sacco di bene anche se non sono sempre lì a dimostrarvelo. Grazie per le seratine tranquille, per le vacanze, per le migliaia di uscite e per essermi stati accanto quando ne avevo più bisogno.

Ringrazio i coinquilini, sono stato fortunato a poter vivere insieme a voi, più o meno a lungo. Conoscervi, parlare di culture diverse, cene, carbonare, intossicazioni, film insieme, le lunghe chiacchierate, le lamentele e per aver condiviso lo stesso tetto, è stato davvero fantastico.

Ringrazio tutte le amicizie fatte all'università, ormai non conto neanche più tutte le persone fantastiche che ho conosciuto. Tutti gli amici del corso, con cui ho condiviso gioie e dolori, esami e giornate di studio intenso. Grazie per tutti quanti i momenti belli e strani che abbiamo passato insieme.

Ringrazio tutte quelle persone che mi sono state accanto nei momenti di sconforto e depressione, ai quali non sono minimamente abituato. Grazie perché sò di poter contare su di voi. Infine ringrazio tutte le persone che ho incontrato, belle o brutte che siano, e che non rientrano nei ringraziamenti precedenti. Grazie per avermi permesso di diventare la persona che sono oggi.

Bibliografia

https://lamenteemeravigliosa.it/laffanno-di-apparire-sui-social-network/

https://www.stefanosalustri.com/blog/7-strumenti-gratuiti-per-fare-scraping/

https://awesomeopensource.com/projects/instagram-scraper

https://www.youtube.com/watch?v=aircAruvnKk

https://www.bnova.it/blog/data-science-qual-e-il-ruolo-del-machine-learning/

https://www.ionos.it/digitalguide/siti-web/programmazione-del-sito-web/che-cose-il-web-scraping/

https://monashdatafluency.github.io/python-web-scraping/section-1-intro-to-web-scraping/#html-dom-or-document-object-model