

**ALMA MATER STUDORIUM
UNIVERSITÀ DI BOLOGNA**

**SCUOLA DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA INFORMATICA**

DISI

Dipartimento di Informatica – Scienza e Ingegneria

**Progettazione e Implementazione di Bot per l'Automatizzazione dei
Processi di Acquisto su Piattaforme e-Commerce Eterogenee**

Tesi in Tecnologie Web T

Presentata da:

CRISTAUDO GIUSEPPE

Relatore:

Prof. BELLAVISTA PAOLO

Anno accademico 2019-2020

Sommario

Introduzione	3
1. Concetti di e-commerce e introduzione agli “sneaker bot”	6
1.1 L’e-commerce	6
1.2 Il brand	8
1.3 Streetwear e sneakers	10
1.4 Hype	14
1.5 Il drop	16
1.6 Caso d’esempio: Supreme	19
1.7 Internet Bot	23
1.8 Sneaker bot	26
1.8.1 Come non essere rilevati	30
1.8.2 Come i sistemi di protezione rilevano e bloccano i bot	31
2. Piattaforme e-commerce e meccanismi “anti-bot”	33
2.1 Piattaforme e-commerce	33
2.1.1 WooCommerce	34
2.1.2 Shopify	35
2.1.3 Magento	36
2.1.4 Demandware	37
2.2 Sistemi di protezione dai bot	37
2.2.1 CloudFlare	38
2.2.2 Captcha	42
2.2.3 Akamai	48
2.2.4 Datadome	50
2.2.5 Variti	51
2.2.6 PerimeterX	53
2.3 Caso d’interesse: Shopify	54
2.3.1 I prodotti	54
2.3.2 Aggiunta al carrello	55
2.3.3 Checkout	56

2.3.4 Sistemi per fermare i bot.....	60
3. Specifiche progettuali.....	62
3.1 Sistema di autenticazione	66
3.2 Sistema di log delle operazioni	67
3.3 Il sistema di notifiche	67
3.3 Modalità “shock drop”	68
3.4 Gestione del pagamento	69
3.5 Le piattaforme supportate	70
3.5.1 Modulo Demandware.....	71
3.5.2 Modulo Magento.....	77
3.5.3 Modulo WooCommerce.....	86
3.5.4 Modulo Shopify	90
4. Tecnologie utilizzate	93
4.1 Python.....	93
4.1.1 Requests	94
4.1.2 BeautifulSoup.....	96
4.1.3 Cloudscraper	98
4.1.6 Discord Webhook	101
4.2 Google Chrome e Fiddler.....	102
4.3 Javascript.....	103
4.4 Performance	105
Conclusione	108
Bibliografia.....	111

Introduzione

Il presente lavoro si propone di descrivere e sviluppare un bot finalizzato a ottimizzare e automatizzare il processo di acquisto sulle principali piattaforme e-commerce utilizzate dai negozi di tutto il mondo. Un bot, per definizione, è un programma in grado di eseguire una serie di operazioni in maniera completamente autonoma al fine di semplificare e velocizzare mansioni poco pratiche per l'essere umano. Per quanto in apparenza possa sembrare accessibile la pratica di acquisto di prodotti via web, in alcuni casi applicazioni di questo genere si rilevano indispensabili. Infatti, per la maggior parte delle persone l'acquisto online si limita solamente a una routine senza interesse. Al contrario, per una nicchia di persone, me compreso, l'acquisto di un particolare prodotto si può tramutare in oggetto di passione, guadagno e studio. Il motivo di tale dedizione e interesse risiede nel modo in cui la cultura dello streetwear e delle sneakers è riuscita a radicarsi all'interno della quotidianità. Secondo il fondatore di Sneaker News, Tu-Ming Wu, *“quando acquisti un capo streetwear, non paghi per il prezzo di abbigliamento più artigianale e di migliore qualità. Stai acquistando una sottocultura, qualcosa di cui sei parte o di cui vuoi far parte”*. La passione per il marchio nasce infatti ben prima del boom tecnologico, quando ancora centinaia di persone passavano intere giornate fuori dai negozi pur di riuscire ad ottenere la sneakers dei loro sogni. Tuttavia, l'avanzamento delle tecnologie e l'evoluzione delle infrastrutture web hanno portato, da una parte i negozi ad adattarsi al modello di vendita online costruendo e-commerce, e dall'altra gli appassionati a trovare il modo di massimizzare le proprie probabilità di acquisto online. I negozi, per accontentare una clientela globale, hanno infatti deciso di rilasciare i propri prodotti basandosi su e-platform già sviluppate, affidandosi a società terze per garantire drop equi tra utenti manuali e utenti che utilizzano bot. Tali società non si limitano dunque a individuare possibili attacchi da parte di bot malevoli, ma si sono specializzate nell'individuare gli “sneaker bot”. Ciò nonostante, la maggior parte dei bot attualmente in commercio

presenta una serie di soluzioni che permettono di eludere i tentativi, da parte dei negozi, di realizzare rilasci di prodotti online in maniera equa.

Riassumendo, i software rendono il processo di acquisto estremamente veloce, generando ancora più esclusività intorno al prodotto e al marchio. Questo effetto domino permetterà ai brand di ottenere una maggiore domanda, instaurando al contempo, concorrenza tra le software house che lavoreranno sempre di più per migliorare i tempi d'acquisto dei propri bot. È proprio all'interno di questo circolo vizioso che due realtà come la moda e la programmazione, prima considerate parallele, si intersecano.

Il software sviluppato si pone dunque come scopo quello di garantire ai propri utenti la possibilità di acquistare i prodotti desiderati sulle piattaforme maggiormente utilizzate e sui siti che sono soliti rilasciare prodotti in edizione limitata. Questo è reso possibile dalla replicazione delle richieste GET e POST effettuate a siti per completare un ordine, affiancata da operazioni di web scraping al fine di ricavare dalle pagine HTML informazioni utili sul prodotto. L'intero sistema è scritto in Python, sfruttando a pieno le sue funzionalità e i moduli a disposizione. L'applicazione verrà sviluppata basandosi su un sistema di autenticazione tramite chiavi univoche al fine di permettere un uso controllato degli utenti. Quest'ultimi saranno in grado di comunicare col sistema tramite una interfaccia a linea di comando intuitiva che rende un'esperienza di utilizzo semplice anche per gli utilizzatori meno esperti in ambito tecnologico. Infine, si prevede un periodo di beta di almeno sei mesi, limitato a pochi utenti fidati, fino al raggiungimento di un adeguato livello di sicurezza sia lato client che lato server di autenticazione e di un numero considerevole di articoli in edizione limitata acquistati con successo durante i rilasci.

Le tematiche appena discusse vengono affrontate nel presente elaborato, che si articola in quattro capitoli, di cui si propone una sintetica descrizione.

Nel primo capitolo sono presentati i principali concetti relativi al commercio elettronico e alla nascita dei rilasci dei prodotti in edizione limitata, citando il famoso

retailer newyorkese “Supreme” come esempio. Si descrivono inoltre i bot su internet, siano essi buoni e maligni, soffermandosi sulla tipologia sviluppata in questo trattato: “SneakerBot”.

Nel capitolo 2 saranno trattate le principali piattaforme e-commerce attualmente utilizzate dai negozi virtuali che vendono articoli streetwear e sneakers, sottolineando le soluzioni a cui quest’ultimi si affidano per garantire drop equi.

Nel capitolo 3 viene introdotto il bot sviluppato, trattando le principali funzionalità e portando alcuni esempi di sviluppo per ognuna delle e-platform trattate.

Infine, nel quarto capitolo vengono illustrate le tecnologie utilizzate per lo sviluppo del software motivando le scelte e le performance raggiunte sui siti supportati.

1. Concetti di e-commerce e introduzione agli “sneaker bot”

1.1 L’e-commerce

Sono gli anni Settanta e per la prima volta si inizia a utilizzare il termine e-commerce per descrivere “lo svolgimento di attività commerciali e di transazioni per via elettronica” [1]. Il primo approccio a questo innovativo sistema di commercio fu in occasione della creazione dell’Electronic Data Interchange (EDI), un sistema volto al trasferimento di ordini d’acquisto e di fatture in formato elettronico. Tuttavia l’adozione dell’EDI da parte delle imprese non riscosse un grande successo a causa degli ingenti costi dovuti alla necessità di creare e mantenere una linea dedicata che unisse i partners delle transazioni. Successivamente, grazie allo sviluppo tecnologico, si sono venute a definire diverse tipologie di e-commerce, ognuna di esse pensata per assecondare al meglio il ruolo delle rispettive controparti commerciali. Tra i vari scenari possibili l’interesse ricade tuttavia sul Business to Consumer o più comunemente chiamato B2C: tipologia di e-commerce nata nel 1979 e utilizzata nel processo di vendita diretta di prodotti e servizi tra un’impresa e un consumatore.

Nonostante lo sviluppo e l’espansione delle tecnologie web il commercio elettronico si poneva lo stesso scopo, distribuire beni e servizi cambiando però il mezzo di comunicazione, Internet in questo caso. Gli esempi di tale tipologia di distribuzione sono svariati, e si estendono da colossi come Amazon e Yoox fino ad attività commerciali più piccole che grazie al web riescono a soddisfare le esigenze di un pubblico sempre più ampio. Dal modello descritto si può evincere come gli attori che operano in una transazione virtuale siano gli stessi di una modalità tradizionale, variando esclusivamente i mezzi con i quali questi comunicano e trattano tra di loro. Sono infatti presenti un venditore (attività commerciale), una piattaforma di vendita proprietaria (e-commerce in questo caso) come Shopify, Magento o WooCommerce e infine un compratore (cliente privato). Il punto cardine dell’intera transazione risiede

dunque nella piattaforma online utilizzata che, come sostenuto nello scritto “Strategy and New Economics of Information” [2], gioca un ruolo di intermediario. I due autori, Evans e Wurstel, sottolineano l’importanza di tali piattaforme spiegando i tre fattori chiave che portano l’e-commerce ad avere un enorme impatto sull’economia.

- Raggiungibilità, ovvero la possibilità di interagire con un numero di clienti potenzialmente illimitato, infrangendo le barriere geografiche mediante Internet.
- Ricchezza, grazie alle sempre più interattive tecnologie web è possibile offrire al cliente un’interazione unica con il prodotto che si vuole acquistare, arricchendo l’articolo con descrizioni, prezzi, foto e disponibilità. In questo modo il cliente ultimo è pienamente informato su ciò che intende acquistare.
- Affiliazione, consiste in un’ottima forma di pubblicità per i negozi e-commerce in cui l’affiliato vende articoli per conto di un venditore a un tasso concordato o percentuale sulla vendita.

L’importanza nell’utilizzo di un negozio e-commerce risiede nel miglioramento dell’efficienza di un’impresa attraverso la rete. In particolar modo l’e-commerce può essere visto come un potente strumento di marketing per ottenere una maggiore reattività, utilizzando Internet come mezzo per proporre i propri prodotti e dialogare con i potenziali acquirenti. I fattori che hanno permesso l’espansione delle vendite online sono molteplici e spaziano dalla possibilità di poter acquistare a qualsiasi ora della giornata in qualsiasi giorno dell’anno, alla possibilità di confrontare i prezzi in modo da valutare attentamente quale sia la piattaforma più conveniente. In questo modo, non solo i compratori godono dei vantaggi generati dall’utilizzo di un e-commerce, ma anche le attività commerciali possono sfruttare la visibilità offerta dalla rete per allargare i propri confini di vendita. Aumentano così le vendite grazie alla possibilità della piattaforma di non dover mai chiudere, di poter risparmiare con lo snellimento della burocrazia e infine di personalizzare l’esperienza d’acquisto online. Quest’ultima gioca inoltre un ruolo fondamentale in quanto, a differenza di un negozio

fisico in cui l'esperienza d'acquisto si basa sugli allestimenti dei prodotti, le luci e gli elementi in vetrina, un negozio virtuale ha la possibilità di creare un'esperienza personalizzata per ogni cliente, generando messaggi e offerte promozionali al fine di invogliarlo all'acquisto. Infine la versatilità della vendita online, rispetto alla tradizionale vendita al dettaglio, ha aiutato il compratore ad appassionarsi al marchio permettendogli di diventare parte integrante della società come descritto di seguito.

Mentre 1989 Tim Berners-Lee, durante i suoi studi presso il CERN di Ginevra, proponeva il progetto per un ipertesto globale conosciuto come World Wide Web, Michael Jordan, 5 anni prima, firmò il suo primo accordo con Nike gettando così le fondamenta del collezionismo di scarpe da ginnastica, le "sneakers". La capacità del cestista dei Chicago Bulls permisero a Nike prima, e Jordan Brand dopo, di instaurare relazioni esclusive e indimenticabili con i consumatori.

1.2 Il brand

Facendo ora un passo indietro, secondo la definizione dell'American Marketing Association, la marca o brand è "un nome, un termine, un simbolo, un design o una combinazione di questi elementi che identifica i beni o i servizi di un venditore differenziandoli da quelli dei concorrenti" [3]. Il processo di branding consiste dunque nel differenziare i prodotti creando così un'esperienza di marca. L'obiettivo finale è far aderire i consumatori al sistema dei valori proposti dal marchio, in una mentalità che sia in grado di creare complicità e intimità tra brand e clienti. La saturazione di offerta e comunicazione rende tuttavia sempre più ardua la diversificazione. Quindi è necessario identificare gli elementi coerenti nel tempo che contribuiscano a rendere l'identità di un brand autentica e unica. Secondo Joachimsthaler e Aaker, *"l'identità di marca dovrebbe contribuire a stabilire un rapporto durevole tra marca e cliente generando un ammontare di valore che, almeno potenzialmente, implica benefici di*

tipo funzionale, emotivo e di auto-espressione e conferendo credibilità a marche garantite” [4]. L’identità che viene comunicata e l’attività di promozione devono dunque coordinarsi in modo tale che l’immagine desiderata dal produttore coincida con quella percepita dal consumatore.

Il consumatore della moda viene in seguito raggruppato in base al sesso, all’età e al reddito disponibile in modo tale che un’attività commerciale possa analizzare il mercato valutando un target ben definito a cui rivolgere la propria offerta. Tuttavia tale suddivisione è stata messa in discussione negli ultimi anni preferendo invece una segmentazione basata su variabili prettamente psicologiche piuttosto che demografiche, come la personalità del consumatore. Una suddivisione alternativa riguarda la “segmentazione tribale” [5]: una metodologia di raggruppamento di individui in cui sono costoro ad aggregarsi spontaneamente sulla base di passioni e interessi comuni. Esempificazioni di questi gruppi possono essere appassionati di basketball oppure chi pratica lo skateboard.

La segmentazione per tribù gode di una forte influenza della diffusione del Word Wide Web, il quale offre infinite possibilità ai gruppi di aggregarsi e alle imprese di comunicare e offrirsi alle tribù distribuite in tutto il globo. Negli ultimi tempi, con l’esplosione delle vendite online, anche il consumo è diventato motivo di costruzione di un’identità e di aggregazione sociale al punto di creare le così dette “comunità di marca”. Un caso particolare riguarda gli appassionati di Supreme, brand e retailer (che verrà trattato successivamente come caso che mi ha portato allo sviluppo di tale progetto), i quali sono disposti a lunghe attese fuori dal negozio pur di entrare in possesso di un capo esclusivo.

Tale pratica non deve risultare estranea alle generazioni del nuovo millennio, come sostiene lo psicologo del consumatore, presso l’University College di Londra, Dimitrios Tsivrikos: *“i Millenials cercano di ispirare o impressionare coloro che condividono lo stesso tipo di interessi che riconosceranno quella particolare T-shirt,*

un gruppo molto piccolo di persone. In alcuni aspetti, ciò che rende le persone vogliose di acquistare un determinato brand è l'aspetto sociale e competitivo: essere in grado di uscire in pubblico indossando vestiti che pochi indossano" [6].

1.3 Streetwear e sneakers

Definire il termine streetwear o il termine street culture non risulta un compito facile, data la loro complessità e combinazione di influenze da parte di diverse culture. Analizzando l'etimologia delle parole "street" e "culture" risulta facilmente intuibile come, senza ombra di dubbio, il termine comprenda una serie di sottoculture di nicchia che prendono forma "nella strada". Molte persone tendono ad etichettare lo streetwear come un prodotto derivante esclusivamente dall'hip hop, tuttavia questa non è una definizione appropriata.

Con certezza si può affermare che lo streetwear prese forma dalle nicchie che praticavano surf e skateboarding agli inizi dei primi anni Ottanta. Il primo grande brand che si fece spazio tra queste nicchie fu Stüssy, ora considerato il padre fondatore dello streetwear conosciuto ai giorni nostri. Shawn Stussy, proprietario del brand Stüssy, diede così un volto a una sottocultura, influenzando altri brand come ad esempio Supreme. Tuttavia, l'attitudine della "cultura di nicchia" su cui è fondato lo streetwear viene persa con l'avvento di Internet e della globalizzazione, la quale porta tale culto ad una serie di nuovi compratori che non credono realmente nel brand, non sviluppano legami con quest'ultimo per ciò che rappresenta ma comprano vestiti e sneakers solamente per lo status quo che questi rappresentano.

Secondo il fondatore di Sneaker News, Tu-Ming Wu, *"quando acquisti un capo streetwear, non paghi per il prezzo di abbigliamento più artigianale e di migliore qualità. Stai acquistando una sottocultura, qualcosa di cui sei parte o di cui vuoi far parte"* [7].

All'interno del settore dello streetwear, i consumatori amano davvero i brand, sono guidati da un radicato senso di appartenenza. Un marchio di successo dovrebbe in tal modo avere una sorta di personalità. Questa personalità umanizza un oggetto inanimato ed è in grado di creare un collegamento con il cliente. La cultura streetwear e l'amore verso essa coesistono in armonia con il mondo delle sneakers.

Le sneakers si sono evolute dall'essere puramente funzionali all'avere un seguito culturale proprio come lo streetwear. Da quando la rivoluzione industriale rese la gomma facilmente reperibile e la produzione di scarpe da ginnastica economica ed efficiente, le sneakers divennero sempre più popolari. Una particolare sneaker che trascese tutte le altre in termini di popolarità e diffusione fu però la Converse. Oltre ad essere stata la prima sneaker approvata e indossata da una celebrità (Chuck Taylor nel 1935), essa fu la prima sneaker ufficiale del team di basketball americano nelle olimpiadi del 1936. Come risultato di questi due eventi, il marchio Converse venne, da quel momento in poi, associato al basketball, dai team di scuole superiori, fino ai professionisti. Gli attori di Hollywood contribuirono notevolmente a rendere popolare questa sneaker soprattutto quando James Dean fu fotografato mentre vestiva Converse bianche, jeans e una maglietta abbinata. L'abbigliamento divenne di conseguenza un simbolo di ribellione e di gioventù per tutti i teenager del periodo. Sebbene alla fine degli anni '70 il modello si trasformò in un misto tra una scarpa sportiva ed una casual, la classica Converse divenne un'icona culturale che tutt'ora viene indossata in modi distinti e personali. Quando si esplora la storia delle sneaker, un altro modello di cui tenere conto è quello della casa tedesca, adidas, dei fratelli Dassler. adidas esplose nel panorama hip hop americano raggiungendo così il podio hall of fame delle sneaker. Grazie al successo del famoso gruppo hip hop RUN DMC il marchio adidas rese popolare la Superstar, scarpa in gomma con dettagli bianchi e neri. Nel libro "Sneakers", l'autore Heard afferma infatti *"DMC trovarono una voce che parlava per un'intera generazione"* e *"DMC ha avuto un sostanziale e a lungo termine effetto sulla cultura giovanile, ma più importante fu DMC a consapevolizzare i marchi della portata dell'affetto che i giovani avevano per i propri mentori"* [8]. adidas risultò

cruciale non solo per il coinvolgimento della musica del momento e le scarpe, ma anche per avere risvegliato le grandi aziende di sneakers. Queste imprese realizzarono finalmente cosa fosse popolare e cosa la gioventù vestisse trovando così un nuovo target di clienti. Dopo l'introduzione di nuove tecnologie performanti per gli atleti, le sneakers cominciarono a essere approvate da celebrità, come Michael Jordan, sviluppando un design adatto per sneakers in edizione limitata. Questo fu un punto cruciale per l'industria in quanto le scarpe da ginnastica cominciarono a essere considerate speciali e la sottocultura che ne derivò nacque con questo concetto.



Fig 1.1 The Air Jordan I and Adidas Superstar on display. Image via Jonathan Dorado

Questi cambiamenti sono aspetti cruciali sia per come si è evoluta la cultura di nicchia delle sneaker sia per il significato che ha assunto il collezionarle. Anche senza includere questa piccola nicchia, le sneakers sono di gran lunga il modello al momento più popolare tra le calzature. Ci sono centinaia, se non migliaia, di modelli facilmente

acquistabili, ricche di stile ma soprattutto sempre disponibili. Per i collezionisti più incalliti questa tipologia di sneaker sono troppo comuni. Gli esperti di sneakers sostengono che “un misto fra cultura popolare, nostalgia, tecnologia e il reddito a disposizione ha pilotato questo fascino verso le sneaker”. (Skidmore)

Per la maggior parte delle persone aspettare fuori da un negozio per tre o più giorni al fine di essere il primo per un rilascio di una sneaker, risulterà da pazzi ma per gli sneakerhead, questa viene definita dedizione. Il possedere un oggetto che nessun altro ha, il vantarsi del proprio stato all'interno della propria nicchia, rende il mondo delle sneaker così attraente. Ogni mese sul sito SneakerNation uno sneakerhead viene intervistato. Le domande spesso includono il nominare il migliore acquisto mai fatto, la loro migliore sneaker di tutti i tempi e soprattutto perché le sneaker rappresentino una così alta priorità. Dopo la revisione di dozzine di interviste è stato facile riscontrare come tutti siano sulla stessa linea quando si tratta di rispondere all'ultimo quesito. Questi collezionisti sono cresciuti in un momento in cui tutte le sneaker erano uguali, e ora che sono diventate un mezzo di espressione di chi sei puoi facilmente affermare la tua identità in un mondo in cui tutto può sembrare uguale. “Puoi dire così tanto di una persona da quello che ha ai piedi” (Andre Speed), “Potresti non avere l'abito più alla moda ma se hai le sneakers, otterrai rispetto” (Skidmore) sono alcune delle citazioni più significative estrapolate dalle loro interviste. In ogni caso, tutti gli intervistati concordano che non esiste sensazione migliore di quando cammini per strada e tutti ammirano le tue sneakers. Come ogni altro tipo di collezione, ogni pezzo ha una storia dietro: da dove viene e come è stato possibile ottenerlo. Un segno in particolare del passato che alla fine crea ricordi. In conclusione, l'enorme mole di appassionati ha aiutato il mercato delle sneaker a manifestarsi con differenti venature. La crescita dei negozi online e i siti di aste hanno fornito ai collezionisti nuovi metodi per trovare le paia più rare. I negozi di sneaker si sono così adattati a questa crescita utilizzando nuovi metodi per la vendita che spaziano dai rilasci first come first served ai più nuovi sistemi di lotteria dove i vincitori sono scelti casualmente realizzando in questo modo un metodo più equo per il rilascio di edizioni limitate.

1.4 Hype

Esiste una stretta relazione tra bisogni psicologici e obiettivi da perseguire. I consumatori tendono a idealizzare e realizzare obiettivi che coincidono coi propri bisogni psicologici. L'economista Thorstein Veblen sostiene, infatti, che

“il consumo può essere basato sulla volontà di pagare un prezzo premium per un articolo al fine di essere considerati una persona più prestigiosa di altri” [9].

A tale affermazione Dholokia e Talukdar aggiungono che i consumatori hanno una tendenza a “imitare il comportamento d'acquisto” [10] degli individui all'interno del gruppo a cui vogliono appartenere, fenomeno osservabile nel consumo di prodotti streetwear e nel mercato dell'abbigliamento di lusso. I consumatori compreranno, dunque, solo ciò che sanno che esiste ed è ben consolidato dalla società; di conseguenza la loro consapevolezza aumenta, traducendosi in un incremento di acquisto compulsivo. Detto questo, quando si verifica un aumento delle vendite l'intenzione di acquisto dei consumatori diminuisce. Ciò si verifica quando il brand smette di essere considerato raro, o meglio esclusivo, per cui l'interesse dei consumatori diminuirà secondo un principio che prende il nome di “Rarity Principle”. Secondo Veblen, infatti, uno dei motivi che spinge un avventore ad acquistare è un confronto individuale che distingue il consumatore dagli altri.

Dalle precedenti considerazioni, possiamo introdurre dunque il concetto di “hype”: quando un determinato prodotto o brand, grazie a consumatori o media, si trova al centro dell'attenzione

In “hype a mathematical equation” [11], l'hype viene definito dalla formula:

$$\text{hype} = \text{notorietà} * (\text{mistero} + \text{esclusività})$$

Nonostante tale formula non abbia nessun fondamento scientifico è in grado, comunque, di rappresentare il moderno mercato dello streetwear.

“L’hype non è necessariamente una cosa buona o cattiva, ma probabilmente rappresenta il più potente strumento che un brand nel 2013 possa usare per vendere merce. Infatti, nessun brand nel 2013 può essere di successo senza una buona dose di hype che lo sostenga.” [12]

La notorietà nella formula gioca un ruolo chiave e prende il significato attribuitogli dalla presa di consapevolezza di Dubois e Paternaults, che si riflette nel quantitativo di percezione dei consumatori a proposito del brand dalla prospettiva del consumatore. Ciò che costituisce la notorietà è l’incognita del brand, come quest’ultimo opera e infine le sue origini. L’esclusività ha un notevole impatto sulla notorietà ed è possibile ricondurla al “Rarity Principle”, trattato in precedenza. Infatti, affinché un prodotto parli di sé è necessario che sia riconoscibile dagli altri e allo stesso tempo costituisca un punto di distinzione. Tale distinzione, come sostenuto da Dubois e Paternaults, potrebbe notevolmente incrementare l’intenzione di acquisto di un consumatore [13]. Infatti, i due sociologi credevano fortemente che un incremento delle vendite diminuisse l’interesse e l’aspirazione per il brand, perché quest’ultimo non seguirà più il “Rarity Principle” e di conseguenza non trasmetterà ai clienti un senso di unicità e considerazione come ci si aspetterebbe. Questo è dunque il motivo per cui l’esclusività e l’alone di mistero aiutano a bilanciare l’interesse e l’effettivo acquisto di un prodotto brandizzato.

Questo paragrafo tende a spiegare come tale accanimento verso il brand e la necessità sempre maggiore di aggiornare il proprio guardaroba, al fine di distinguersi dagli altri individui appartenenti alla stessa “tribù”, abbia portato allo sviluppo di software in modo da aumentare le possibilità di acquistare determinati prodotti a tiratura limitata

dai negozi e-commerce.

1.5 Il drop

Una metodologia che va sempre più ad affermarsi per i brand è quella di alimentare la domanda realizzando prodotti in edizione limitata i quali, per scelta aziendale, vengono dunque prodotti in quantità inferiore alla domanda o per un lasso di tempo limitato. Grazie al mercato creato dall'incontro dell'offerta e della domanda, un quantitativo limitato di prodotti comporta un aumento dell'esclusività, creando un certo senso di urgenza. Il fondatore di Supreme, James Jebbia, con una celebre frase disse: “Se so che ne posso vendere 600, allora ne produco 400” [14]. I prezzi più richiesti spesso si esauriscono quasi all'istante. Basti pensare a come i prodotti più esclusivi del brand newyorkese si esauriscano sul sito online in tempi che spesso si aggirano sotto i dieci secondi. Come si può facilmente intuire tali tempistiche sono spesso inarrivabili per un utente manuale se si parla di trovare un prodotto, aggiungerlo al carrello, inserire i propri dati e infine completare il pagamento. Questo rispecchia una realtà in cui software per automatizzare il processo di acquisto sono sempre più comuni e necessari. Tali programmi rendono il processo di acquisto estremamente veloce, generando ancora più esclusività intorno al prodotto e al marchio che otterrà una maggiore domanda portando concorrenza tra le software house che lavoreranno sempre di più al fine di migliorare i tempi d'acquisto dei propri software andando a creare un circolo vizioso che unisce sempre di più due realtà prima considerate parallele: la moda e la programmazione.

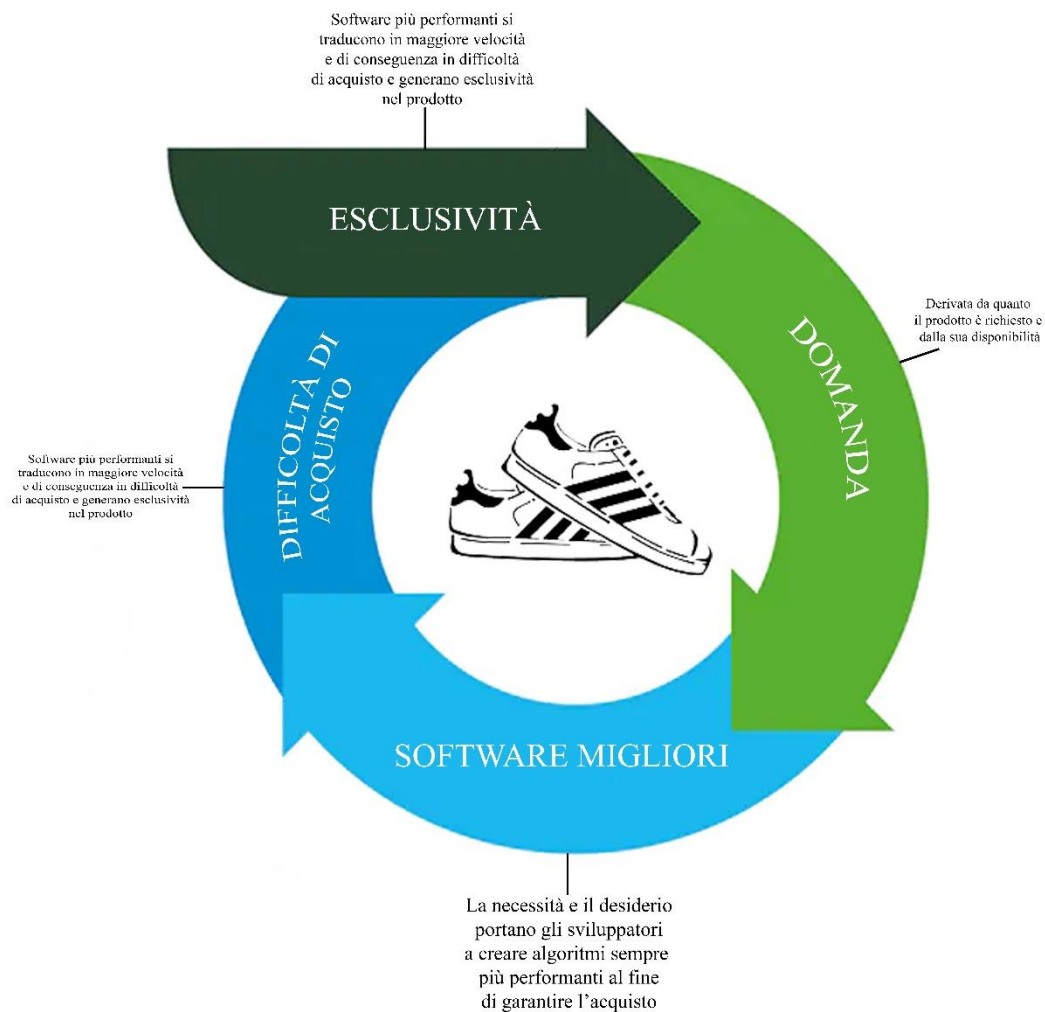


Fig. 1.2 Schema esplicativo del legame fra sneaker botting e esclusività

Questo appeal per i consumatori di linee esclusive e in edizione limitata è semplice: compra ora o perdi per sempre “one-change-to-buy”. L’amministratore delegato di Vetements, Guram Gvasalia, afferma *“il lusso è come un appuntamento”, “se qualcosa è disponibile ed è di fronte a te, è meno desiderabile. La scarsità è ciò che lo definisce. Uno dei modi per creare scarsità è ridurre la curva di offerta. Più domanda c’è, maggiore è il desiderio che si crea”* [15].

Secondo il professore Adam Alter: *“se vuoi creare frenesia attraverso la scarsità e il tuo marchio ha un forte seguito, tutto ciò che devi fare è rilasciare un gran numero di prodotti diversi, ma molto pochi di ciascuno di essi. La bellezza della moda è che è visibile; tutti sanno cosa stai indossando, quindi se indossi un prodotto raro e alla*

moda ottieni il beneficio di sapere da te stesso che possiedi qualcosa di scarso, ma anche il vantaggio di essere in grado di mostrare ad altre persone che possiedi un prodotto raro” [16].

Il rilascio controllato di nuovi prodotti in edizione limitata prende il nome di drop, il quale viene realizzato con frequenza settimanale o mensile, ad un ritmo notevolmente più veloce del tradizionale ciclo della moda per mantenere alto l’interesse dei compratori. La tipologia di drop è parte integrante della “shopping experience” del consumatore, introducendo un valore aggiunto attraverso il quale il marchio rende unica e memorabile l’esperienza d’acquisto. Secondo Cova, Giordano e Pallera i drop rappresentano una strategia di gestione degli eventi, alternativa affermata negli ultimi anni e riconducibile ai principi del cosiddetto “guerrilla marketing”: metodo che consiste nel dare vita ad azioni coinvolgenti ed eclatanti in grado di attivare un passaparola virale tra le persone per ottenere una risonanza elevata con investimenti contenuti [17].

Il concetto di drop è stato sperimentato inizialmente in Giappone dal designer Hiroshi Fujiwara e reso popolare dai marchi streetwear come Supreme e Vetements che hanno adottato questo sistema sia nei negozi fisici che in quelli online.

Supreme rappresenta dunque un chiaro esempio del successo di tale metodologia, che ogni giovedì mattina vede lunghissime file di persone attendere pazientemente il proprio turno al fine di aggiudicarsi un pezzo del drop settimanale. Questo accade in tutti e dodici negozi Supreme distribuiti in tutto il globo sottolineando come il fenomeno non sia definito da limiti geografici e come possa contagiare qualsiasi individuo indipendentemente dalla propria cultura. Oltre alle code che si formano davanti ai negozi, Supreme offre ai propri amanti del brand la possibilità di acquistare l’ultimo prodotto esclusivo mediante il proprio e-commerce, basato sulla piattaforma Shopify. Tali prodotti vengono venduti in modalità “first come first served”, causando in questo modo la necessità al collezionista di doversi adattare, utilizzando determinati

programmi al fine di raggiungere il proprio obiettivo: l'acquisto del prodotto. Tali programmi generano un'enorme mole di richieste al sito nel momento del drop; infatti, secondo quanto dichiara il responsabile dell'e-commerce di Supreme, Samuel Spitzer, ogni giorno dopo il rilascio di un drop il traffico sul sito web si impenna con punte del +16.800% [18]. Per marcare l'efficacia di questa strategia basti pensare come l'8 dicembre del 2017, il giorno del rilascio del prodotto più esclusivo del brand newyorkese, la box logo, il sito ricevette 986.335.133 visualizzazioni e 1.935.195.305 tentativi di acquisto.

Quando un'impresa limita la disponibilità di un articolo molto popolare, essa può trasformare in esperienza il semplice fatto di possedere questo bene. Facendo scarseggiare i propri prodotti, Supreme sublima l'esperienza del possederne uno.

1.6 Caso d'esempio: Supreme

Supreme viene fondata a New York City da James Jebbia nel 1994, su Lafayette Street, nel cuore di Manhattan. La storia e il seguito del brand sono strettamente legati al primo store interamente basato e pensato per la cultura Hip Hop e la nicchia degli skater. L'idea di aprire il negozio nacque nel periodo in cui Jebbia lavorò presso il negozio di Shawn Stussy in Inghilterra, osservando come il marchio Stüssy stesse sempre più spopolando.

James Jebbia decise dunque di abbandonare il proprio lavoro e di dedicarsi completamente all'apertura della propria attività a New York City in cui non erano ancora presenti degli "skate shop" adeguati a soddisfare l'elevata domanda del mercato di skateboard newyorkese [19].

“Il negozio è stato in grado di diventare il santo Graal della cultura giovanile di strada, mescolando icone della città: moda, musica e politica all’interno delle mura e immediatamente ...” [20]

“The store was able to become the holy grail of high youth street culture by curating a mix of the city’s iconography – fashion, music, celebrity, and politics within its walls and then instantly sledge-hammering the city’s high-low playing.” [21]

Molto presto, Supreme divenne la casa della cultura skate a New York. I prodotti che includevano semplicemente il design del logo messo su una felpa o su una maglietta divennero velocemente l’emblema dello skateboard in città. Più che un negozio, Supreme sembrava quasi una galleria d’arte. Ross Wilson, uno dei più grandi fan del brand newyorkese da quasi due decenni, disse *“Entrando, il negozio era ben illuminato con pareti bianche luminose e un soffitto molto alto. L’ambiente pulito del negozio e i prodotti esposti in modo impeccabile si presentavano più come una boutique di alta moda e l’incenso di Nag Champa che bruciava all’entrata mescolato al debole aroma di erba proveniente dalla piccola stanza dietro il registratore di cassa, diede al posto un odore distinto”*. [22]

Con lo svilupparsi della cultura underground non furono solo gli skater a bramare l’ultimo articolo del marchio. A 10 anni di distanza dall’apertura del primo negozio James Jebbia decise dunque di aprire un nuovo negozio a Los Angeles in modo da riuscire a soddisfare anche i fan della costa ovest statunitense. Tuttavia per mantenersi al passo con la continua e crescente domanda da parte del sempre più ampio pubblico, nel 2006 Supreme lanciò il proprio negozio online. Il sito ottenne un immediato successo per i nuovi fans del marchio, tutti i clienti in giro per il mondo potevano finalmente fare parte del culto dietro il famoso negozio di skate.

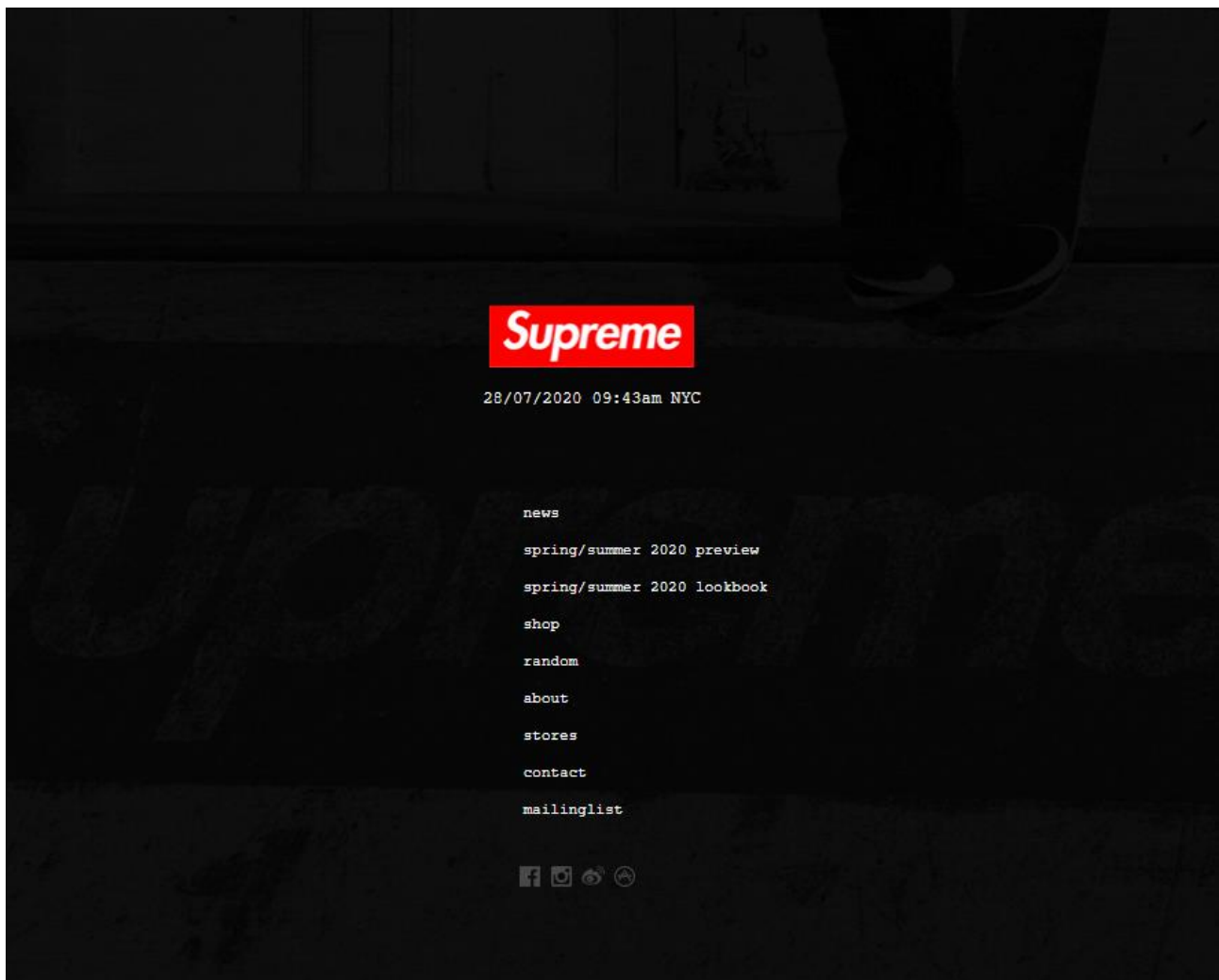


Fig. 1.3 Pagina principale <https://www.supremenewyork.com/>

Con il nuovo sito funzionante i drop di Supreme divennero sempre più difficili da gestire. Le lunghe file formatesi fuori dai negozi erano ora accompagnate da bot, trasformando i giorni dei drop in un evento focale del calendario settimanale nella scena streetwear. Molte più persone non riuscirono ad avere ciò che desideravano a causa dei quantitativi limitati di Supreme e la continua crescita di domanda. Tuttavia durante il corso degli ultimi anni Supreme ha tentato di bloccare tali software in grado di acquistare prodotti Supreme, in tempi decisamente minori di un normale utente manuale. Il primo approccio utilizzato fu la scelta dell'introduzione del sistema re CAPTCHA di Google al momento dell'acquisto; tale soluzione, attualmente ancora in uso, utilizza un'azienda di terze parti al fine di verificare l'identità "umana" dell'utente

tramite la risoluzione di un determinato puzzle. Tuttavia tale soluzione non risulta troppo affidabile in quanto per molti anni all'interno l'e-commerce americano permetteva il bypass. Nel settembre del 2017 Supreme introduce Tohru, un sistema per monitorare i movimenti del mouse al fine di riconoscere la presenza umana. Anche questa soluzione fu facilmente annientata da parte degli sviluppatori di bot semplicemente navigando sul sito Supreme tramite mobile. Durante il 2018, un primo vero tentativo di combattere la presenza di software automatizzati fu quello di inserire dei caratteri speciali tra i nomi degli articoli. In questo modo, nel momento della ricerca tramite caratteri dei prodotti, questi ultimi non venivano rilevati in quanto la corrispondenza con le parole chiavi non era garantita come ad esempio utilizzando la B del normale alfabeto oppure la “B” dell'alfabeto cirillico. Anche questo ostacolo venne risolto dagli sviluppatori semplicemente traslando i caratteri di altri alfabeti nei caratteri comuni. Sempre nel 2018 Supreme introduce all'interno del proprio codice HTML degli script JavaScript che rilevano specifiche parole chiavi appartenenti allo sviluppo di bot come ad esempio “delay” o “autofill”. Anche in questo caso la soluzione risultava ovvia. Infine, verso il concludersi del 2018 viene introdotta una nuova potente contromisura che prende il nome di “Pooky”. Il Pooky non è altro che uno script realizzato in JavaScript che viene eseguita in maniera automatica quando si visita il sito di Supreme. Tale script procede a generare una serie di cookies i quali saranno inviati al server al momento della conferma dell'ordine, nel caso siano validi l'ordine andrà a buon fine. I bot developer dovranno dunque effettuare un reverse engineering al fine di ottenere una versione valida dello script criptato al fine di ricreare cookies validi che verranno dunque inviati in fase finali per far credere al sito supreme che l'ordine sia stato fatto manualmente [23].

1.7 Internet Bot

Dai paragrafi precedenti si intuisce come il desiderio di ottenere il prodotto abbia portato, a chi ne avesse le facoltà, a sviluppare bot in grado di acquistare in modo autonomo. Un bot (abbreviazione di robot) in terminologia informatica è un programma che accede alla rete attraverso lo stesso tipo di canali utilizzati dagli utenti umani (può per esempio accedere alle pagine Web, inviare messaggi in una chat). Programmi di questo tipo sono diffusi in relazione a diversi servizi di rete, con scopi vari, ma sempre legati all'automazione di compiti troppo gravosi o complessi per utenti manuali [24].

Nel caso di interazioni con Internet, si parla di Internet Bot. Tale categoria comprende tutte le applicazioni software che eseguono operazioni (scripts) in maniera autonoma su Internet. Tipicamente, i bot compiono operazioni semplici e ripetitive molto più velocemente di quanto una persona possa fare. La struttura base di un bot comprende:

- Logic application. L'eseguibile, un codice leggibile dalla macchina che lo sviluppatore di bot scrive e il dispositivo esegue.
- Database. Una collezione di informazioni che il bot utilizza in modo tale da sapere che operazioni eseguire. Un bot è in grado di salvare informazioni addizionali sul proprio database, come quando un web scraping bot scarica contenuti da un sito web.
- API, permette ai bot di utilizzare funzionalità esterne senza che lo sviluppatore necessiti di scriverle [25].

Più della metà del traffico web è generato da bot. Questi dati derivano da uno studio effettuato da Imperva Incapsula sulla base di più di 16,7 miliardi di visite a 100mila domini selezionati in modo casuale. Come mostra il grafico sottostante, nel 2016 la quantità di traffico derivante dai bot è stato del 51,8% (tra bot “buoni” e “cattivi”), contro il 48,2% di quello reale [26].

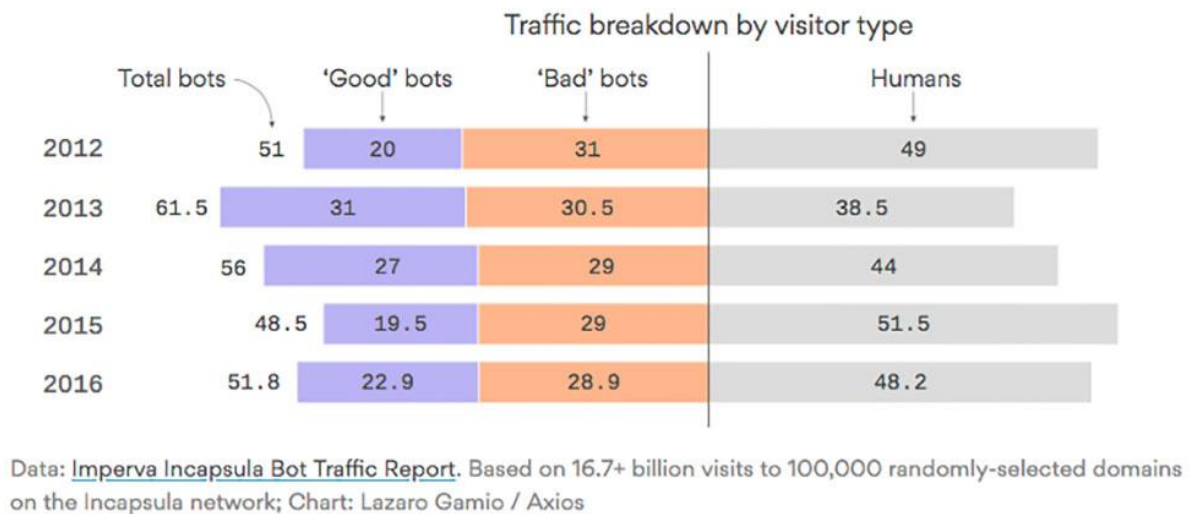


Fig. 1.4 Imperva Incapsula Bot Traffic report

Il traffico web generato dai bot è rintracciabile utilizzando una serie di parametri tra cui:

- Tendenze di traffico: picchi anormali di traffico potrebbero indicare che bot stanno colpendo il sito. Questo risulta particolarmente vero quando il traffico si verifica in orari insoliti.
- Frequenza di picchi: insoliti alti e bassi potrebbero essere un segno di bot malevoli. Ad esempio, i bot che colpiscono una determinata pagina e successivamente cambiano indirizzo IP, appariranno come un 100% di picco.
- Fonti di traffico: durante un attacco, il canale principale che invia traffico diretto registrerà la presenza di nuovi utenti e nuove sessioni.
- Prestazioni del server: un rallentamento in termini di prestazioni del server potrebbe essere un segno di uso di bot.

- Indirizzi IP e fonti di linguaggio sospetti: un incremento di attività da parte di un intervallo di indirizzi IP sconosciuti o da una regione con cui di solito non si interagisce. In modo analogo avviene per lingue poco utilizzate.

Dalla loro natura, è possibile distinguere i bot “buoni” e quelli “malevoli”. Un uso malevolo dei bot è la coordinazione e l’esecuzione di un attacco autonomo a server internet, come ad esempio il denial of service attack. Questa categoria comprende:

- Spambot, software programmati per inviare spam. Tali script tipicamente creano un ingente numero di account usati successivamente per inviare messaggi di spam. Un esempio è rappresentato dagli e-mail spambots, i quali ricercano indirizzi e-mail da materiale online al fine di creare una lista di e-mail a cui inviare messaggi di spam.
- Bot per attacchi DDoS. L’impossibilità di erogare il servizio si verifica quando è presente un overload nelle componenti singole di una infrastruttura online. Se questo fenomeno è deliberatamente verificato da un attore esterno, si è in presenza di un attacco DDoS. Per eseguire questo tipo di attacco, l’attaccante inonda il sistema con un numero estremamente elevato di richieste da diversi dispositivi che combinati formano una gigantesca rete di bot, conosciuta anche come botnet. Come risultato, il sistema non sarà in grado di rispondere in tempo o di processare le richieste.
- Scraping attack bot. Questa tipologia di bot preleva contenuti dai siti web. Essi sono responsabili di esaminare database, estrapolare informazioni e duplicarle su un altro sito. Spesso le imprese eseguono scraping al fine di collezionare informazioni a proposito del proprio sito web e dei propri utenti. In questo caso, la società stessa sviluppa il bot e gli dà accesso al sistema. Contrariamente, tale tipologia di bot malevoli, ruba informazioni da terze parti per poi venderli o compiere operazioni non autorizzate [27][28].

Al contrario, la parte buona comprende tutti quei bot che eseguono operazioni utili al fine di garantire la miglior esperienza utente possibile. Spesso l’attività di questi bot è

supervisionata dalle regole contenute nel file “robots.txt”. In questo caso la categoria è composta per la maggior parte da ChatBot, algoritmi in grado di fornire risposte a quesiti posti da utenti o più in generale di interagire con quest’ultimi.

In conclusione, è possibile catalogare il software sviluppato in questo elaborato come un bot buono. Infatti, nonostante utilizzi un numero elevato di richieste il suo obiettivo non è quello di impossibilitare o rallentare i servizi offerti di un server come nel caso di bot sviluppati per attacchi DDoS. Non si tratta nemmeno di Scraping attack bot, in quanto l’attività di scraping della pagina web o del database si limita ad estrapolare informazioni per l’acquisto del prodotto e non a immagazzinarle per secondi fini.

1.8 Sneaker bot

L’universo streetwear o più nello specifico delle sneakers è un vero e proprio business e, dove vanno i soldi i bot seguono. Questo motiva fortemente gli sviluppatori di bot che di continuo cercano di eludere i sistemi di rivelazione.

Come suggerisce il nome, gli sneaker bot hanno come obiettivo principale l’acquisto di sneaker, ovvero scarpe da ginnastica a tiratura limitata. Successivamente verrà descritto il mondo delle sneaker e dello streetwear sottolineando alcune soluzioni che i proprietari di e-commerce creano al fine di garantire un’ottima esperienza d’acquisto per i clienti umani. Di seguito si riporta a titolo d’esempio il rilascio di una adidas Yeezy, sneaker realizzata dalla casa tedesca in collaborazione con l’artista Kanye West.

Per la loro natura, i rilasci a tiratura limitata sono brutali in termini di traffico web. Quando una sneaker popolare è messa in vendita il traffico cresce in maniera esponenziale in un arco di tempo decisamente breve in quanto bot e clienti umani cercheranno di ordinare l’oggetto desiderato. Il seguente grafico che riporta il numero

di richieste nel tempo (traffico complessivo e traffico derivante da bot) durante il rilascio di una Adidas Yeezy, una sneaker edizione limitata, sui server Datadome, piattaforma di protezione informatica trattata nello specifico nel capitolo successivo.

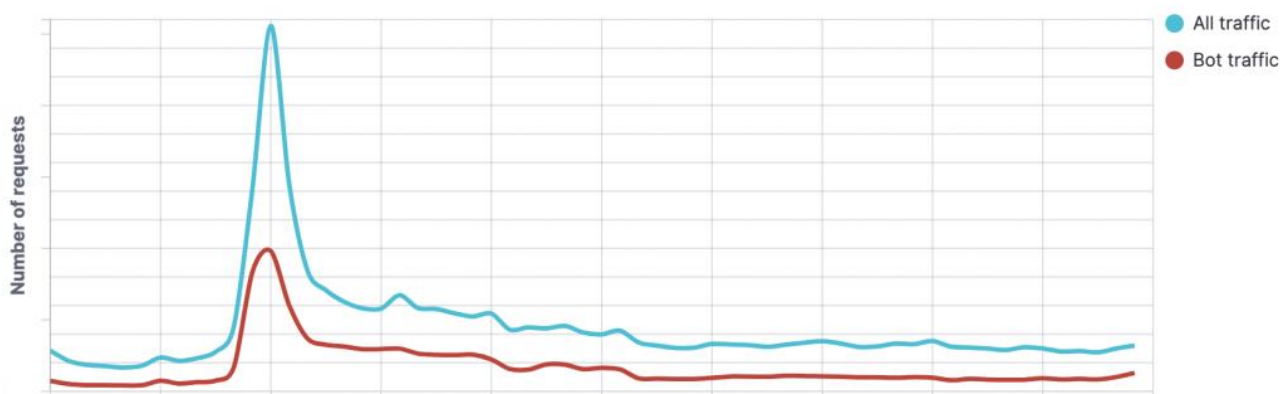


Fig. 1.5 Grafico del numero di richieste nel tempo durante il rilascio di una Adidas Yeezy

Il picco di traffico si ha quando viene aperta la vendita. Una volta che il prodotto sarà esaurito il traffico torna velocemente alla normalità. Mentre l'infrastruttura tipica di una attività commerciale è abbastanza scalabile al fine di gestire un numero prevedibile di picchi come durante i grandi lanci, il triplicare del traffico legittimo sommerebbe costi aggiuntivi che si potrebbero facilmente evitare. Nel sito in questione, infatti, il traffico non raggiunge mai il server dal momento che Datadome, sistema di protezione bot utilizzato dal sito web adidas, blocca la maggior parte dei bot già dalla prima richiesta. Inoltre, poichè i bot vengono bloccati dall'effettuare acquisti, i reali clienti del sito possono competere in termini equi per acquistare la sneaker più desiderata.

L'obiettivo di questa tipologia di bot è quella di automatizzare il processo di acquisto di sneakers a tiratura limitata. Tali software hanno due grossi vantaggi rispetto agli acquirenti manuali: la velocità e la possibilità di gestire ed effettuare richieste multiple simultaneamente. Infatti, quando una nuova sneaker viene rilasciata, essa può esaurirsi in una manciata di minuti se non secondi. Per questo, acquistare un paio di sneakers edizione limitata è diventata una pratica quasi impossibile senza l'aiuto di bot. Tra gli

utilizzatori degli sneaker bot, sono presente in piccola parte collezionisti di sneakers, gli “sneakerheads”, i quali desiderano massimizzare le loro possibilità di accaparrarsi un paio delle loro scarpe preferite. Secondo Wikipedia,

“Uno sneakerhead è una persona che collezione, scambia o ammira le sneaker come passione. Uno sneakerhead, potrebbe inoltre avere una grande esperienza nel distinguere sneaker autentica da repliche”. [29]

Più frequentemente gli sneaker bot sono utilizzati per operazioni su larga scala. Persone alla ricerca di profitto traggono vantaggio della loro velocità per acquistare il numero maggiore di scarpe in edizione limitata nel tempo minore possibile che possono poi rivendere, spesso con guadagni significativi. Per chi non conosce questo mondo potrebbe pensare a cifre irrisoria, ma il mercato di rivendita delle sneaker è estremamente remunerativo. Infatti, secondo la piattaforma StockX, il principale mercato di sneakers online, il mercato globale di rivendita delle scarpe da ginnastica è attualmente valutato sei miliardi di dollari. Le sneakers più ambite possono fruttare centinaia se non migliaia di euro al paio, creando un margine di guadagno succulento per il rivenditore. Di conseguenza la concorrenza per il “Santo Graal” delle calzature è agguerrita e i bot si trasformano quindi in strumenti indispensabili per chi vuole partecipare alla partita. Esistono diverse varianti di sneaker bot che passano da browser completamente automatizzati a estensioni per i browser. In ogni caso la maggior parte non richiede alcuna capacità di programmazione da parte dell'utilizzatore, il quale si limita a interagire con il bot tramite interfacce user-friendly o CLI dove possono specificare diversi parametri come, ad esempio, l'URL del prodotto che desiderano acquistare, la o le taglie desiderate e il metodo di pagamento da utilizzare per completare l'ordine come il numero di carta di credito o PayPal. Spesso questi software vengono integrati con piattaforme come Discord, applicazione VoIP, in modo tale che l'utilizzatore possa ricevere notifiche riguardo gli acquisti completati con successo. I migliori bot possono facilmente eludere sistemi anti-bot come, ad esempio, evitare che l'utente risolva i Captcha da solo integrando il bot con l'API di Captcha farm come ad

esempio 2captcha o anti-captcha. Dal momento in cui gli sneaker bot fanno esattamente dove cliccare o effettuare richieste e possono in maniera autonoma riempire i form di spedizione e pagamento con le informazioni che l'utente ha precedentemente inserito, sfrecciando attraverso il processo di completamento dell'ordine molto più velocemente di quanto un uomo possa realisticamente fare. Alcuni di questi bot, inoltre, forniscono applicazioni per il cellulare che possono automaticamente mandare il token 3Dsecure al telefono dell'utente in modo da velocizzare ulteriormente l'intero processo. Proprio come le sneakers più attraenti anche i migliori bot sul mercato hanno un'alta richiesta. Di conseguenza anche la vendita di sneaker bot è diventata un business. Il valore di mercato per la maggior parte dei bot spazia dai 200 ai 400 euro iniziali più una spesa mensile che varia dai 30 ai 50 euro. Ma, come le scarpe più ambite, anche i bot sono venduti in edizione limitata. Comprarli direttamente dagli sviluppatori è diventato estremamente difficile e il mercato di rivendita è esorbitante, con bot che raggiungono cifre dai 4000 ai 10000 euro.

La ragione per cui gli sviluppatori di bot limitano la distribuzione del proprio prodotto è per massimizzare l'efficienza stessa. Infatti, minore è il numero di copie distribuite, minore è il numero di utenti che condividono la stessa impronta digitale, riducendo la possibilità di essere rilevati e minore è il numero di utenti che competono per acquistare la stessa scarpa edizione limitata, incrementando così la possibilità singola di "copping".

"Copping è l'atto di acquisto con successo di un paio di sneaker. In modo indipendente da come avvenga, copping è l'obiettivo finale di qualsiasi sneakerhead in questo mondo. Specialmente se si tratta di sneaker in edizione limitata." [30]

Al posto di aumentare il numero massimo di licenze gli sviluppatori aumentano il proprio guadagno offrendo acquisti supplementari. Molti di loro vendono un ampio ventaglio di prodotti e servizi al fine di massimizzare le possibilità di acquisto degli

utenti. Ad esempio, i bot sono spesso venduti insieme a servizi per acquistare proxy residential o data center, i quali riducono notevolmente il rischio degli sneaker bot di essere rilevati e di conseguenza bloccati. Un altro esempio molto frequente è la vendita di membership di cookgroup privati.

“Un cookgroup è una chat di gruppo, basata su piattaforme come Discord o Slack, dove i migliori sneakerheads, resellers, fanatici di sneaker e collezionisti entrano per discutere di sneaker, prossime uscite, monitorare i rilasci, comprare bot, etc.” [31]

L’obiettivo di tali gruppi è di condividere informazioni confidenziali che aiutino i propri membri ad acquistare con successo le sneakers, come l’orario del prossimo rilascio, i link di particolari scarpe, proxy che funzionano su un determinato sito web o consigli su come evitare di essere rilevati e bloccati.

1.8.1 Come non essere rilevati

Gli sviluppatori di sneaker bot conoscono bene i principali meccanismi di rilevazione e riescono in questo modo ad eluderli. Tra i meccanismi praticati ci sono:

- “Fake browser fingerprints”: i più sofisticati sneaker bot mentono sul browser utilizzato e sulla impronta http, come informazioni reali del browser, user agent, eliminazione della navigazione web driver, etc.
- “Simulating human behavior”: al fine di apparire umani, spesso gli sneaker bot non si pongono come obiettivo quello di acquistare il prodotto nel minor tempo possibile, ma si limitano ad essere un po’ più veloci dei competitor. Invece, mimano realisticamente i movimenti del mouse, eventi generati dal tocco dello schermo o il battere sulla tastiera.
- “Residential IP addresses”: Questi particolari indirizzi IP, vengono utilizzati meno rispetto ad indirizzi derivanti da data center, in questo modo rendono i bot più difficili da essere rilevati;

- “Low request volume per IP address”: Al contrario di bot che mirano a rubare credenziali o a effettuare attacchi DDoS, generalmente gli sneaker bot non generano un grosso volume di richieste. Questo porta il sito web a identificare un comportamento umano tipico in cui vengono effettuate un numero adeguato di richieste dallo stesso indirizzo IP;
- “Bypassing Captcha”: Gli utenti bot sono consigliati a utilizzare diversi account Google con una lunga cronologia e una “reputazione pulita”, in questo modo se Google si fiderà dell’account, esso mostrerà una Captcha più facile che sarà più veloce da risolvere.

1.8.2 Come i sistemi di protezione rilevano e bloccano i bot

Di seguito vengono riportati invece alcuni dei meccanismi che Datadome utilizza per identificare la presenza di bot. Successivamente verranno trattate le principali soluzioni utilizzate delle attività commerciali al fine di mitigare gli utenti bot.

- “Fingerprinting”: attraverso script JavaScript vengono collezionati segnali per identificare i bot che utilizzano headless Chrome, Puppeteer o Selenium;
- “Advance IP reputation analysis”: il sistema Datadome analizza ogni singola richiesta di migliaia di siti web su tutto il globo. Grazie alla loro libreria in tempo reale di informazioni, esso può identificare milioni di indirizzi IP usati comunemente dai bot, siano essi proxy data center o residenziali;
- “Machine learning-based behavioral detection”: attraverso complessi algoritmi basati su intelligenza artificiale è possibile identificare se la richiesta proviene

da un umano o da un bot basandosi su segnali lato client e informazioni sul traffico fra tutti gli utenti.

Una volta rilevato lo sneaker bot, è possibile bloccarlo per l'intera durata della vendita. Inoltre, la protezione offerta da Datadome, include algoritmi specifici di rivelazioni in cui il sito web sia a conoscenza dell'utilizzo o della presenza di uno sneaker bot. [27]

2. Piattaforme e-commerce e meccanismi “anti-bot”

2.1 Piattaforme e-commerce

Una piattaforma e-commerce è un software che permette alle imprese che desiderano digitalizzare il proprio business di creare un negozio online gestendo tutti gli ambiti di cui quest'ultimo è costituito. Tali piattaforme offrono inoltre una serie di strumenti di controllo che permettono alle attività commerciali di personalizzare l'aspetto e lo stile grafico del proprio store online grazie a semplici funzioni “drag and drop” messe a disposizione direttamente dagli sviluppatori. Oltre a funzioni basilari di personalizzazione, queste e-platform, offrono un ampio ventaglio di configurazioni, rendendole, non solo personalizzabili, ma anche aggiornabili e scalabili in maniera tale da rimanere di pari passo con il business, senza diventare obsolete. Pur non esistendo una piattaforma migliore delle altre, ognuna di quelle trattate successivamente eccellerà in ambiti specifici e risolverà problemi peculiari. Queste piattaforme rappresentano il metodo più intuitivo per sviluppare il proprio negozio virtuale, infatti non serve disporre di una squadra di informatici dedicata alla propria attività online poiché l'esperienza di creazione del proprio sito web è estremamente user-friendly ed economica se comparata alle spese di creazione di un sito e-commerce completamente da zero. Un altro punto a favore delle piattaforme e-commerce è rappresentato dalle “Hosted Solution”, le quali permettono di delegare la gestione del sito a terze parti senza doversi preoccupare di aspetti come web hosting, sicurezza o aggiornamenti.

Le e-platform prese in considerazione sono WooCommerce, Magento, Shopify e Demandware in quanto, nonostante non siano tutte molto popolari, rappresentano la quasi totalità delle piattaforme e-commerce utilizzate da attività commerciali solite vendere prodotti in edizione limitata.

Ecommerce Platforms Popularity, 10K Web Stores, September 2017

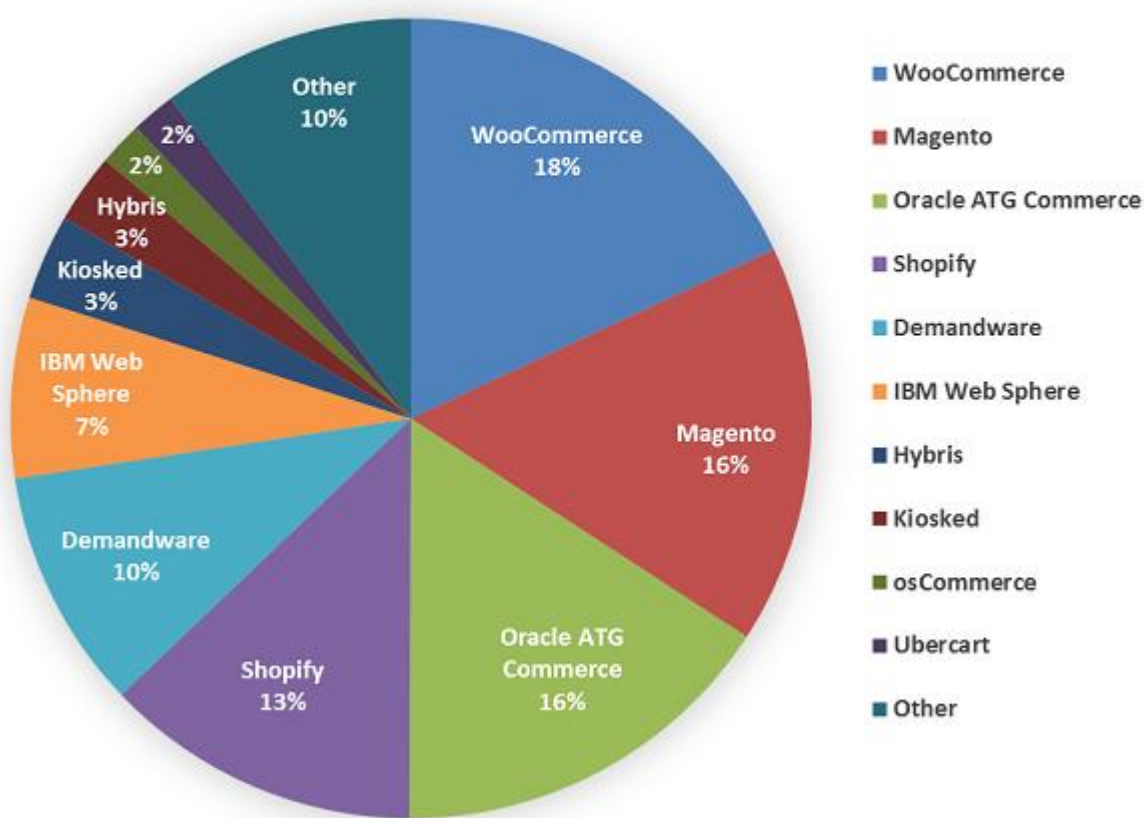


Fig. 2.1 “Who Leads the Market of E-commerce Platforms in 2017?” research by aheadworks.com

2.1.1 WooCommerce

WooCommerce venne rilasciato nel 2011 dai creatori di WooThemes, un'estensione di WordPress che permette di sviluppare una attività online con cui vendere prodotti e servizi. WooCommerce, con 10,8 milioni di download e grazie alla sua facilità di personalizzazione e semplicità d'uso, è attualmente impiegato dal 18% dei negozi online di tutto il mondo divenendo una delle soluzioni migliori per le aziende che gestiscono un piccolo volume di affari. Un altro punto di forza della piattaforma è il suo forte legame con WordPress e alla sua fedele community, attraverso la quale sviluppatori e appassionati possono discutere, o più nello specifico sviluppare una serie di plugin con cui adattare la piattaforma alle proprie esigenze, migliorando

ulteriormente l'esperienza di acquisto e la gestione dell'attività commerciale online. La facilità d'uso nel caso di WooCommerce va però a discapito della sicurezza, infatti questa piattaforma necessita di costanti aggiornamenti di sicurezza al fine di risolvere le falle di sistema che permettono agli sviluppatori di bot di effettuare acquisti multipli e aggirare i meccanismi di "bot protection" messi a disposizione dalla piattaforma stessa.

2.1.2 Shopify

Shopify nasce nel 2006 e attualmente viene utilizzata dal 9% dei negozi online italiani. Nonostante la percentuale sembri notevolmente minore rispetto alle altre piattaforme, essa rappresenta la quasi totalità degli e-commerce che negli Stati Uniti provvedono a rilasciare in maniera regolare prodotti in edizione limitata. Analizzando infatti le tendenze di Google riguardo le piattaforme e-commerce si può notare come Shopify sia di gran lunga una delle piattaforme a livello globale più utilizzate.



Fig. 2.2 Google Trends, confronto in base all'interesse tra le piattaforme e-commerce trattate

La popolarità di Shopify non deriva solo dalla sua facilità d'uso, ma anche dalla miriade di template disponibili che rendono il proprio sito accattivante e moderno, invogliando, in questo modo, il cliente a comprare. Tale piattaforma offre diverse tipologie di piani di abbonamento alle imprese che vogliono affacciarsi al mondo online, in modo tale

che quest'ultime possano gestire a pieno la personalizzazione del proprio sito. Anche l'assistenza gioca un ruolo fondamentale in Shopify, se si hanno problemi con la piattaforma è sempre disponibile un servizio clienti che varia dalla live chat al supporto Twitter.

Shopify verrà descritto successivamente come caso d'interesse, analizzando l'inventario dei prodotti, le API e le diverse protezioni che la piattaforma mette a disposizione alle attività commerciali al fine di combattere l'utilizzo di bot.

2.1.3 Magento

Sviluppato nel 2008, Magento vanta più di duecentocinquanta mila negozi virtuali, in particolare modo viene utilizzato dal 25% dei negozi online stilati nella lista dei cento mila siti e-commerce più cliccati. Magento costituisce la migliore piattaforma dedicata ai professionisti della vendita online con maggiore esperienza nello sviluppo di applicazioni web. La notorietà di tale piattaforma è attribuibile principalmente alla sua estrema modularità ed estensibilità, che le permettono, nonostante gli elevati costi, di essere utilizzato dalla maggior parte dei migliori e-commerce. Nonostante venga considerato uno dei Content Management System (software per chi aiuta a creare, gestire e modificare i contenuti di un sito) più completi nel settore, la scarsa popolarità è dovuta principalmente ai costi e alla complessità. Magento rappresenta una soluzione “all-in-one” per gli e-commerce: offre una vasta gamma di funzionalità e permette il download e l'installazione sul proprio server oppure di utilizzare il servizio hosted tramite il quale è possibile delegare diverse attività direttamente alla piattaforma. In entrambi i casi, la piattaforma offre una ottima integrazione con il servizio CloudFlare, hosting solution in grado di fornire diversi gradi di protezione da bot.

2.1.4 Demandware

Demandware, acquistata da Salesforce nel 2016, permette la realizzazione di una e-platform basata su cloud e completamente integrata con soluzioni mobile, intelligenza artificiale e una serie di servizi relativi alle attività B2B e B2C. Il principale punto di forza di tale piattaforma sono i continui aggiornamenti a cui è sottoposta, non a caso è una tra le prime soluzioni a supportare il sistema di pagamento ApplePay rendendo l'intera esperienza di acquisto ancora più semplice per i clienti. La sua integrazione con la piattaforma di intelligenza artificiale proprietaria, Salesforce Einstein, permette attraverso l'analisi dei clienti, di offrire loro consigli personalizzati su prodotti, risultati di ricerca e altro ancora. Esempi di negozi che utilizzano tale piattaforma sono adidas, YeezySupply oppure tra i siti supportati dal software qui presentato Awlab e Susi.

2.2 Sistemi di protezione dai bot

Gartner, multinazionale specializzata in consulenza strategica, ricerca e analisi nel campo della tecnologia stima che *“entro il 2023, oltre il 30% delle applicazioni web per interazioni pubbliche saranno protette da servizi WAAP (Web Application and API Protection) sul cloud, che forniscono una combinazione di protezione contro gli attacchi DDoS (Distributed Denial of Service), mitigazione dei bot, protezione delle API e soluzioni WAF. SI tratta di un incremento a partire da meno del 10% come si attesta oggi. [33]”*

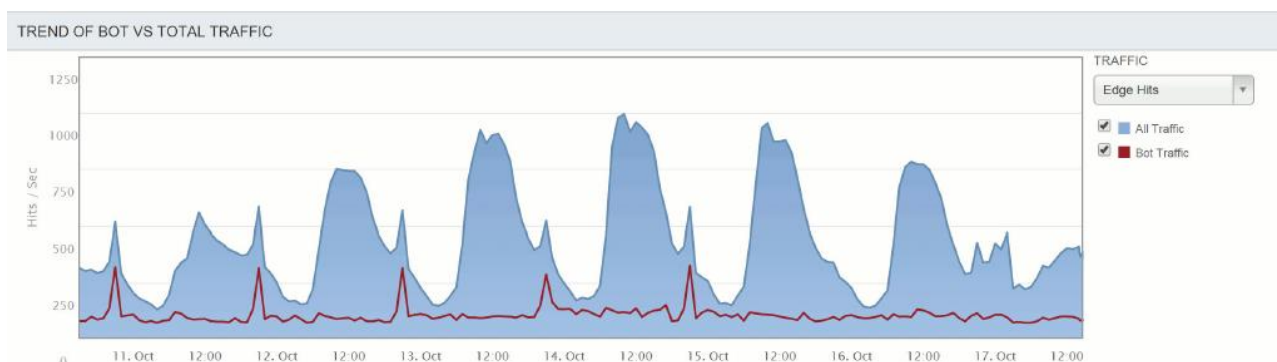


Fig. 2.3 Tendenze del traffico reale a confronto con il traffico generato da bot

Come trattato nel capitolo precedente, il sempre più morboso interesse verso il mondo dello streetwear e dei suoi prodotti a tiratura limitata, ha portato i clienti interessati ad adattarsi alle nuove modalità di rilascio dei negozi. Se fino a qualche anno fa i fan più sfegatati del marchio si mettevano in fila giorni prima del rilascio del prodotto gestito mediante addetti alla sicurezza, nella modalità di drop degli e-commerce la sicurezza è spesso lasciata ad aziende di terze parti che proveranno a garantire un rilascio del prodotto desiderato equo per tutti gli utenti.

Ogni attività commerciale che prevede tali rilasci può valutare se utilizzare tali sistemi di protezione o meno. Spesso però, i proprietari di questi negozi, sono affezionati al marchio e al prodotto che rilasciano tanto quanto la miriade di fan che andranno poi a tentare di acquistarlo; per questo motivo saranno decisamente motivati ad utilizzare una o più soluzioni al fine di bloccare tentativi di acquisti multipli o di bot, dando così la stessa possibilità agli utenti manuali. Nei paragrafi che seguono verranno trattati i sistemi e le aziende che forniscono servizi di protezione dai bot agli e-commerce soffermandosi su eventuali soluzioni al fine di eluderli.

2.2.1 CloudFlare

CloudFlare viene fondato nel 2009 da Matthew Prince, Lee Holloway e Michelle Zatlyn e lanciato successivamente nel settembre del 2010 alla “TechCrunch Disrupt

conference”. Essenzialmente si tratta di una compagnia di sicurezza per siti e di una infrastruttura web che fornisce una serie di servizi tra cui Content Delivery Network (CDN), DDoS mitigation, sicurezza a livello di rete e un servizio di Domain Name Server (DNS). CloudFlare risiede tra il sito e-commerce e i visitatori, comportandosi come un reverse proxy che fornisce sicurezza, prestazioni e velocità al sito.

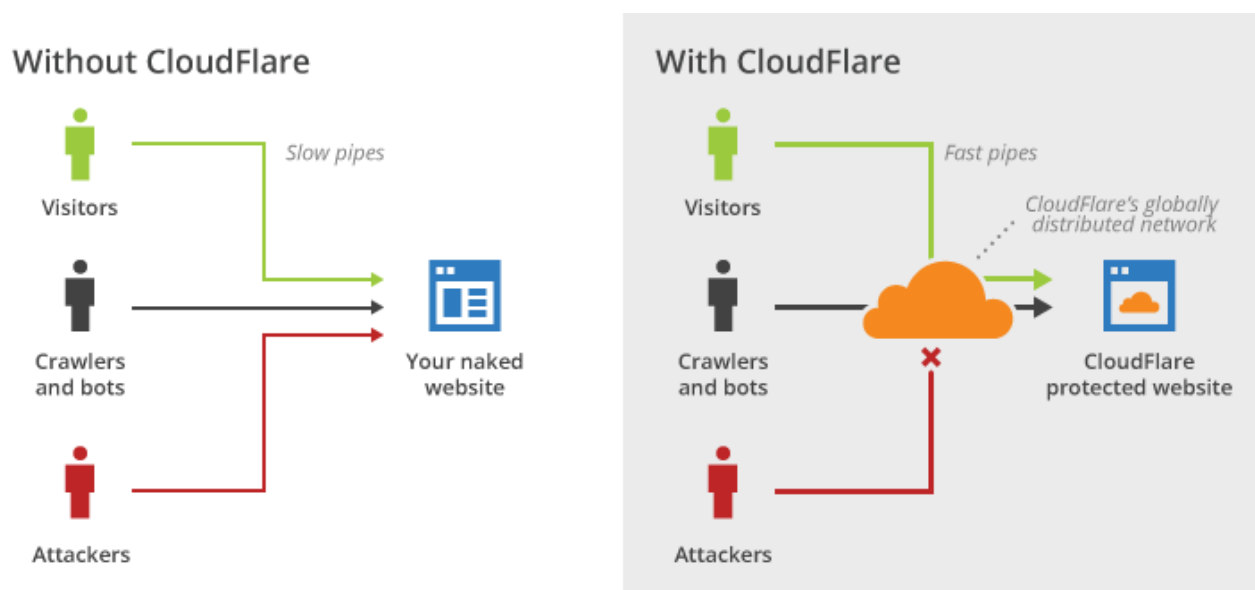


Fig. 2.4 Schema di funzionamento di CloudFlare

Più in generale, quando un sito è stato impostato per usufruire dei servizi di CloudFlare, il suo traffico è direzionato verso il sistema network proprietario invece che essere direttamente servito al visitatore. Questo permette all’infrastruttura di ottimizzare il sito in modo da garantire tempi di caricamento più veloci e una migliore performance, resa possibile da una minimizzazione del codice HTML, CSS, Javascript e dall’utilizzo del CDN. Il CDN crea diverse copie del sito salvandone ognuna in cache collocate nei trentacinque data center distribuiti globalmente permettendo che vengano restituite al visitatore in base alla posizione geografica da cui quest’ultimo accede. Un altro vantaggio riguarda il livello di sicurezza. Esso permette di proteggere un sito web da attacchi come il “distributed denial of service” (DDoS) o SQL Injection e XSS JavaScript Injection utilizzando un servizio di Web Application Firewall (WAF),

oppure come nel caso di interesse, a limitare l'impatto dei bot in modo da garantire un accesso più facile a un utente manuale, risparmiando notevolmente anche la larghezza di banda a disposizione e le risorse del server.

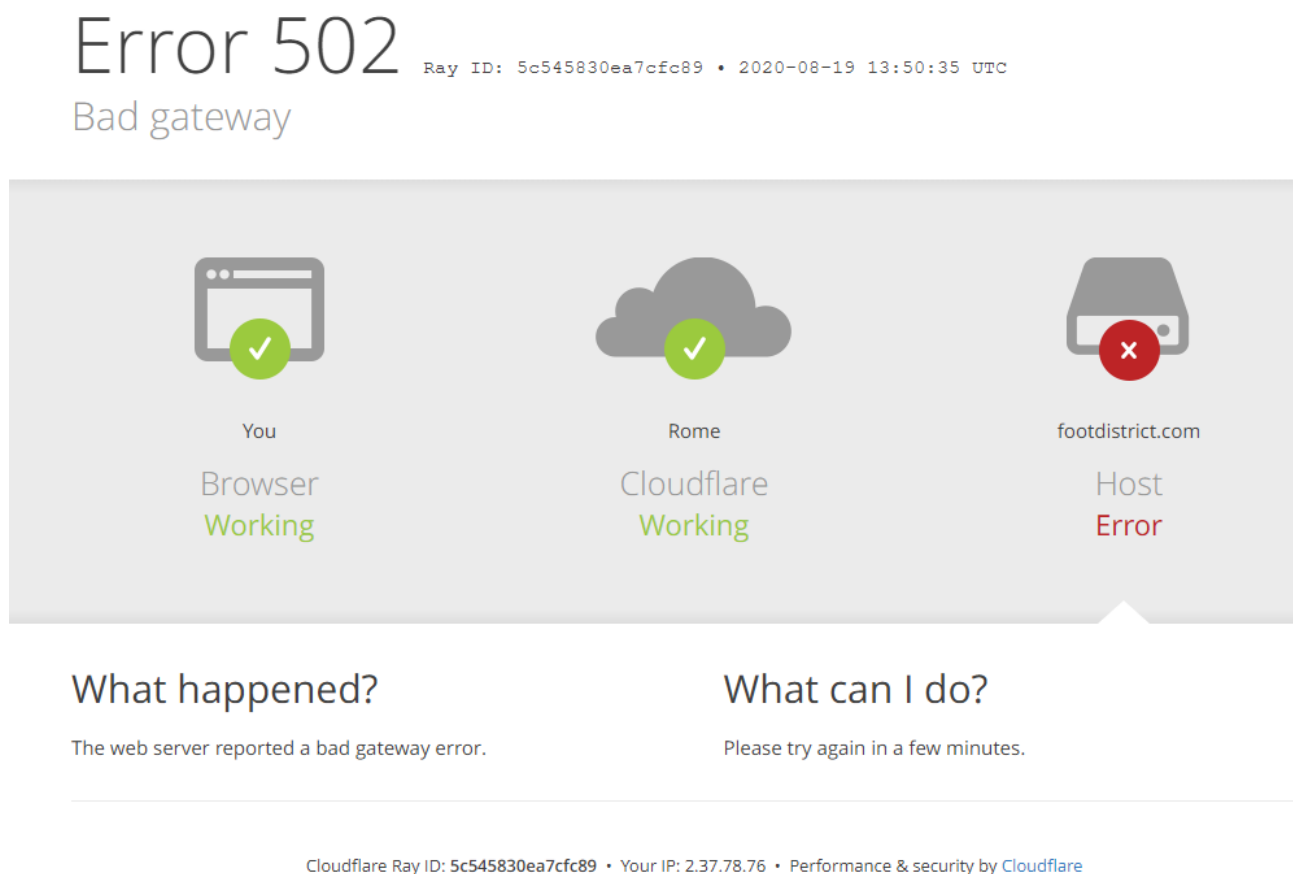


Fig. 2.5 Pagina restituita da CloudFlare quando l'host non è disponibile durante un drop

CloudFlare può essere integrato con qualsiasi e-platform, rendendolo notevolmente versatile. Per configurarlo, infatti, non necessita di alcun set up lato server o installazione, bensì basta semplicemente cambiare il dominio DNS del proprio sito web, utilizzare l'indirizzo logico di un server CloudFlare e successivamente configurare quali dei sotto domini si desidera proteggere e migliorare a livello di prestazioni. Grazie alla sua facilità di integrazione, ai suoi benefici in ambito SEO che

permettono di ottenere punti per quanto riguarda l'indicizzazione del proprio sito web, e al suo prezzo, CloudFlare risulta una delle protezioni maggiormente utilizzata dalle attività commerciali al fine di rendere più equo possibile il rilascio di prodotti edizione limitata.

Di seguito viene illustrato come il tentativo di accesso da parte di un utente non comporti una sfida visibile da risolvere come nel caso di Captcha, trattati successivamente. Come esempio verrà utilizzato un noto negozio di Taranto, tra i migliori distributori del marchio adidas in Italia. L'accesso da browser mostrerà la seguente schermata.

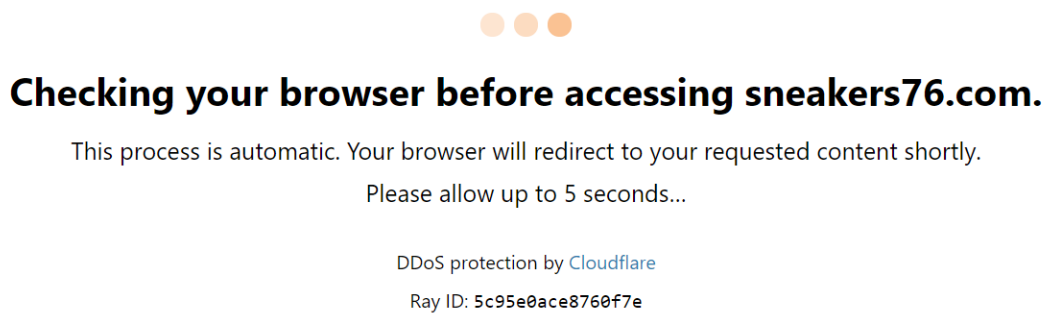


Fig. 2.6 Pagina CloudFlare 'I'm under attack mode'

Al contrario, la richiesta al medesimo indirizzo effettuata da un software non sufficientemente elaborato riporterà la seguente risposta costituita da un insieme di caratteri alfanumerici indecifrabili.

```
b'\x03A\r\x00\xc3\xa4o\xc2\xbf\xc3\x8co\xc2\x98\xc3\xb1\xc3\xa3\xc3\xbf\xc2\x95\xc3\x9d\xc2\x968K\x10s\xc3\xb40\xc2\x91\xc2\xbd/\n\xc3\xbbue\xc3\x88\xc2\x82\xc2\x96\xc2\x8c\xc2\x91T0\xc3\x99\xc3\xb3\x7f\xc3\xb7{=\xc2\xa7\xc2\xaaaf\xc3\x96\xc2\xa9v'\xc3\x8dv!)\n\x02$7\xc3\x89{\x191(^'\xc2\xa1\xc3\xa7\x04K\xc3\xbd\xc3\xa7*\xc3\xa4\xc3\xa4\xc2\x84\xc3\x9c"\xc3\x8e]\n\xc2\xbd\x1a\xc2\xabdc2\x81\xc2\xad\xc2\x87\xc2\x9d\xc3\x81\xc2\xaf>1/\xc2\x92\xc3\xb9\xc3\xa4\xc3\xa8\x03gg8VBVP\xc2\xbd)\n\x07%\xc3\xb5}1' {\xc2\x95r\xc2\x8f\xc3\xa3\x18\xc2\xa5\xc2\x87\xc2\x94\xc3\x8c2\xc3\x917\xc3\xbf;\n\xc2\xbb\xc3\x8e{t\xc3\xa7\xc3\xb6\xc3\x93/\xc3\x8a\xc2\x87\xc2\x8b\xc3\xacV\xc3\xab]\n\xc3\x89>\x19\xc2\x89\xc2\xa71\xc2\xaa!\xc3\x90\xc2\xbc\xc3\x98\xc2\x8c\xc3\x84f\xc2\xa2\xc2\x84\xc3\xbc-gM\xc3\x89s?\xc2\xae}\nI45RB%WU9\x13\x7f\x1a\xc3\x8ccj\xc2\x9e\xc2\xa6\xc3\xa31\xc2\x83\xc2\xb9Fh1\x0c\xc2\x90\xc3\xb8\xc2\x97\xc3\xa6/\n\xc3\x86\xc2\x93Db<-\xc3\x8e\xc2\x8b\xc2\xabc{t\xc2\xb0\xc3\x9f\x1a\x05e\xc2\xba\x1b"\xc2\x9a\xc2\x84\xc3\x98 D\xc3\xa9\x0b\xc3\x89$|\n<\n\xc2\x8a\xc2\x93\x0e\xc2\xaer\x05e\xc3\xb2@P\xc2\x9f\xc2\x94\xc2\xa8vpU&+\xc3\x8e\xc2\x8b5s\xc2\x94\xc3\x98\x082\xc2\x86\xc3\xa6
```

Fig. 2.7 Body della response in seguito ad una GET effettuata sul sito sneakers76.com

In questo caso, la protezione si limita a bloccare tentativi di web scraping, necessari allo sviluppatore di bot al fine di prelevare il codice HTML sorgente della pagina in modo da poter ottenere gli elementi necessari per completare l'ordine. Tali elementi non comprendono solo gli identificativi delle taglie del prodotto, ma anche form key o token utili per convalidare ogni richiesta effettuata al server in cui risiede il sito.

Tuttavia la modalità CloudFlare scelta dal sito è facilmente eludibile semplicemente utilizzando un header ben formattato per eseguire la richiesta alla pagina. Nel caso in cui la sfida imposta da CloudFlare risulti più ardua verrà utilizzata una libreria sviluppata da VeNoMouS, la quale, tramite l'utilizzo di un engine JavaScript, consentirà allo script di impersonare facilmente un normale browser senza offuscare e analizzare esplicitamente il codice Javascript di CloudFlare.

2.2.2 Captcha

Captcha è l'abbreviazione di Completely Automated Public Turing test to tell Computers and Humans Apart, tradotto Test di Turing pubblico e completamente automatico per distinguere computer e umani. In altre parole, si tratta di un sistema di verifica che permette di dimostrare se si sta interagendo con una persona umana o meno. Questi test sono stati sviluppati alla fine degli anni Novanta dal gruppo di ricerca e sviluppo di AltaVista, al fine di impedire ai bot di aggiungere URL al proprio motore di ricerca. Parallelamente, Luis von Ahn e Manuel Blum diffusero una prima idea di test Captcha, intendendo qualunque tipo di algoritmo in grado di distinguere persone da computer. Nella sua forma più classica, un Captcha, prevede l'inserimento da parte di un utente di caratteri alfanumerici leggendo una stringa disturbata da elementi grafici di varia natura al fine di rendere l'operazione di verifica più difficili ad eventuali programmi di riconoscimento.



Fig. 2.8 Captcha, risoluzione tramite interpretazione di una frase

A questa tipologia di Captcha si aggiungono altre sfide che prevedono ad esempio la risoluzione di operazioni rese maggiormente incomprensibili alle macchine tramite la presenza di numeri romani o la dettatura come previsto nei Captcha sonori.

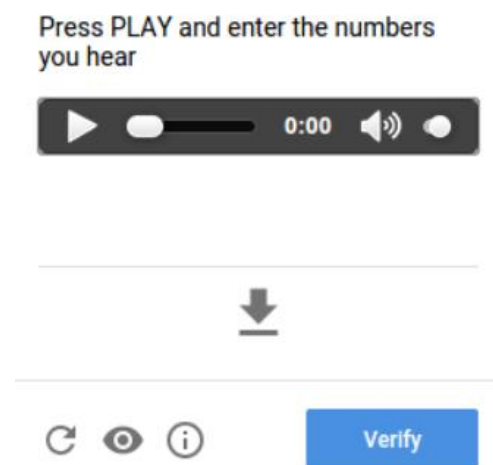


Fig. 29 Captcha, risoluzione tramite interpretazione di suoni

Spesso gli algoritmi sviluppati per realizzare i test vengono divulgati in modo da dimostrare che la sicurezza del metodo non derivi dalla conoscenza di un algoritmo, bensì dalla classificazione “hard” di un problema nel campo dell’intelligenza artificiale.

Per quanto riguarda invece il loro utilizzo da parte dei diversi e-commerce, è bene specificare come essi non vengano utilizzati solo per prevenire grossi spam di azioni, ma anche per garantire la veridicità di diverse azioni durante l'acquisto di un prodotto come possono essere il login, l'aggiunta al carrello o il completamento dell'ordine. In questo modo l'azione eseguita è teoricamente subordinata dalla risposta umana. Tuttavia, l'utilizzo massiccio di Captcha rischia di causare problemi di utilizzo al cliente finale. Il motivo del successo in termini di protezione dai bot deriva dal basso costo e dalla affidabilità che i test Captcha hanno: sono infatti completamente automatici e non richiedono di norma interventi umani per la somministrazione o la manutenzione. Di seguito si riportano due delle soluzioni Captcha più utilizzate dagli e-commerce legati al rilascio di prodotti a tiratura limitata.

- **reCaptcha**

reCaptcha è un servizio di sicurezza offerto da Google, nato per proteggere i siti web da bot. La maggior parte dei siti utilizza ancora reCaptcha V2, lanciato nel 2014. Nel caso in cui il comportamento di un visitatore risulti sospetto, reCaptcha V2 mostrerà una sfida che il visitatore dovrà necessariamente risolvere al fine di provare la propria natura. La difficoltà della sfida dipende da quanto Google ritenga chi interagisce con la pagina realmente umano, e varia dal cliccare il bottone con scritto "Non sono un robot" alla risoluzione di una sfida che contiene immagini o audio.

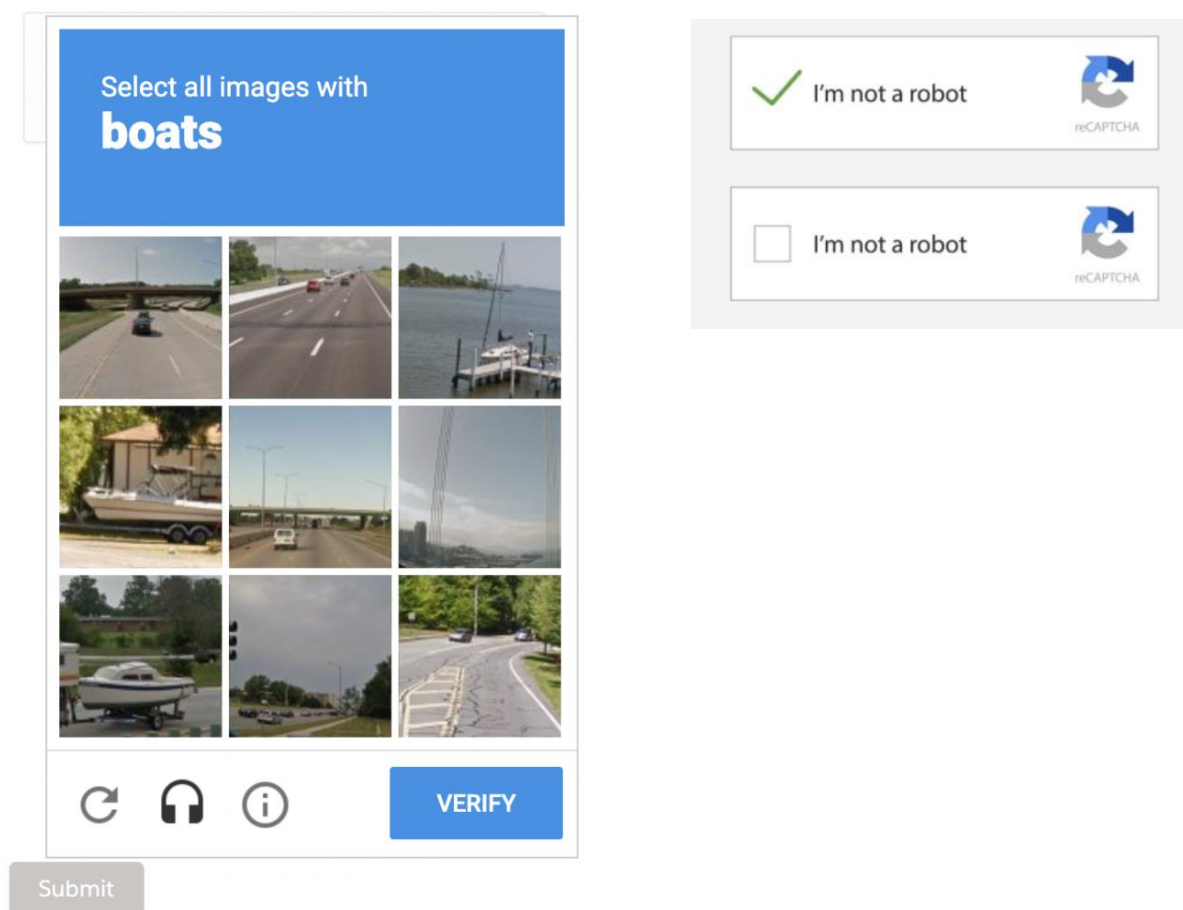


Fig. 2.10 Esempio di Google reCaptcha

Il sistema reCaptcha V2 è infatti sviluppato su “advanced risk analysis system”, un algoritmo nativo basato sui cookie di Google. Questo porta un vantaggio in termini di affidabilità a chi utilizza ad esempio Google Chrome oppure a chi ha eseguito un accesso con un account Google. Tale meccanismo è anche alla base di reCaptcha V3 sviluppato in modo da fornire una maggiore user experience, essendo essenzialmente trasparente per i visitatori di un sito web. Non è presente alcuna sfida da risolvere, bensì reCaptcha V3 monitora di continuo il comportamento di un visitatore in modo da determinare se si tratta di un umano o di un bot. Per ogni richiesta che un utente esegue, reCaptcha V3 restituisce un punteggio da 0 a 1 che rappresenta quanto possibilmente la richiesta è stata effettuata da un bot: se è vicina allo 0 essa è stata eseguita da un bot, in caso contrario è possibile che la richiesta sia stata effettuata da un umano.

```

{
  "success": true,
  "hostname": "recaptcha-demo.appspot.com",
  "challenge_ts": "2020-08-15T14:00:46Z",
  "apk_package_name": null,
  "score": 0.9,
  "action": "examples/v3scores",
  "error-codes": []
}

```

Fig. 2.11 Punteggio reCaptcha V3 consultabile all'indirizzo <https://recaptcha-demo.appspot.com/recaptcha-v3-request-scores.php>

In questo modo il sistema migliora l'esperienza utente eliminando in gran parte il numero di sfide necessarie a verificare la propria identità, complicando però il lavoro dell'amministratore web, il quale dovrà impostare per ogni azione possibilmente a rischio la soglia che farà scatenare la sfida. Uno dei problemi che porta reCaptcha V3 è la larga mole di informazioni che deve utilizzare al fine di effettuare una decisione corretta del comportamento.

Per quanto possa sembrare complesso il sistema offerto da Google per verificare l'identità dell'utente, la protezione che esso assicura può essere considerata superficiale. Infatti, nel caso si fosse in possesso di un account Google con elevato punteggio, a questo non verrà chiesta alcuna sfida e il bot potrà proseguire nel processo d'acquisto senza alcun intoppo. Nel caso di reCaptcha V2 è persino possibile utilizzare una serie di algoritmi basati su intelligenza artificiale in grado di risolvere in maniera autonoma la sfida. Un'altra modalità spesso utilizzata è quello di integrare nel proprio bot un "manual captcha harvester", un modulo che permette di prelevare la sfida che viene mostrata e farla risolvere manualmente in modo tale che chi sta utilizzando il bot debba esclusivamente preoccuparsi di risolvere le sfide, mentre il bot procede ad aggiungere al carrello e successivamente completare l'ordine. Infine, un altro metodo per risolvere il reCaptcha prevede l'utilizzo delle cosiddette Captcha farms, società di terze parti che si occupano di risolvere manualmente le sfide presentate in cambio di un costo minimo. Esempi sono i siti anti-captcha e

2captcha i quali con una spesa che varia da uno a tre dollari possono risolvere fino a 1000 reCaptcha V2.

- hCaptcha

Alla fine del 2018, Google rilasciò la sua ultima versione del rinomato sistema di rivelazione dei bot. Se da una parte reCaptcha V3 viene considerato un ottimo rilevatore, molti ritengono che quest'ultimo violi alcuni concetti di privacy. Nasce così hCaptcha (dove la "h" sta per human), un servizio di identificazione della natura dell'utente che in modo analogo al sistema competitor chiede di etichettare una serie di immagini. Tuttavia, invece che mostrare immagini proprietarie di Google, utilizza immagini fornite da aziende specializzate in intelligenza artificiale, le quali utilizzano le informazioni ricavate dalla risoluzione della sfida come database per i propri algoritmi.

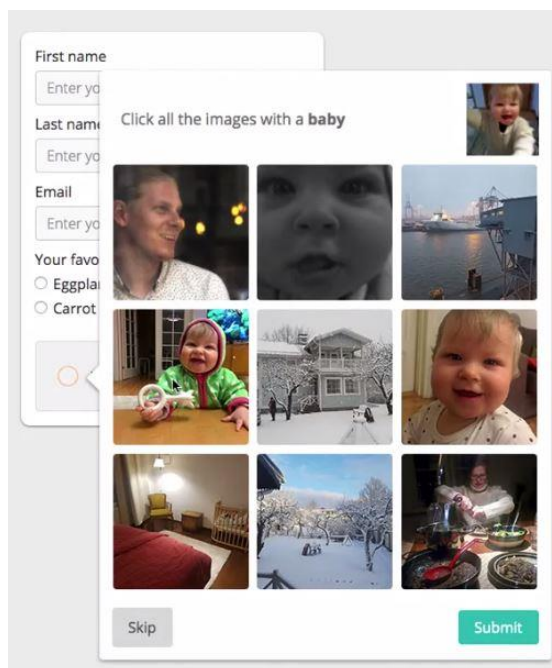


Fig. 2.12 Esempio di captcha proprietario di hCaptcha

Questo comporta che, mentre Google richiede un costo per utilizzare il proprio sistema, hCaptcha aiuta sia il sito web a proteggersi dai bot, sia gli sviluppatori di algoritmi basati su intelligenza artificiale rendendo questo processo, non solo

gratuito per chi lo utilizza ma anche remunerativo. Il punto di forza di hCaptcha ha spinto il colosso di protezione dei bot CloudFlare a implementare hCaptcha piuttosto di reCaptcha per le proprie applicazioni. Il CEO di CloudFlare, Matthew Prince in una intervista sostiene:

“We choose reCaptcha because it was effective, could scale, and was offered for free which – was important since so many of Cloudflare’s costumers’ user our free service.” [35]

In termini di contrasto ai bot, hCaptcha risulta essere più promettente rispetto ai competitor, in quanto attualmente non sono presenti strumenti per eludere la sfida e l’unico modo per dimostrare la propria identità umana consiste nel risolvere la sfida proposta.

2.2.3 Akamai

“Akamai primeggia sulla concorrenza grazie alle sue solide funzionalità di segnalazione e risposta agli attacchi”. [36]

Akamai Technologies, Inc. fondata nel 1998 da Daniel Lewin è un’azienda che fornisce una piattaforma per la distribuzione di contenuti via Internet. Questa e-platform garantisce protezione dalle minacce informatiche e performance di alto livello effettuando una copia dei contenuti del sito sui propri server. L’intero sistema di protezione dai bot è basato sul rinomato sistema proprietario Bot Manager integrato alla Akamai Intelligent Platform. In questo modo gli utenti reali e i bot si collegano al sito web tramite il server Akamai e il Bot Manager rileva, identifica e gestisce i bot prima che raggiungano le applicazioni o i data center.

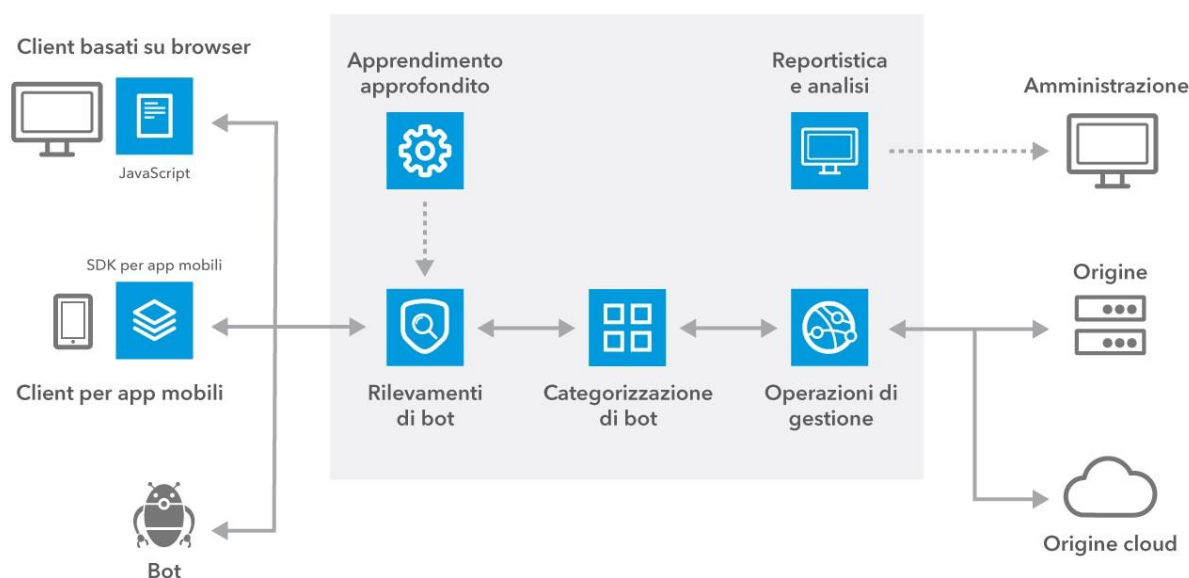


Fig. 2.13 Schema Akamai Bot Manager

Bot Manager fornisce dunque un framework flessibile per gestire al meglio i diversi bot che ogni giorno tentano di accedere ai siti web identificando le richieste non appena arrivano, suddividendole in gruppi e infine applicando la politica di gestione più appropriata per ciascuna categoria. La rilevazione dei comportamenti sospetti in questo caso è data da una combinazione di fattori che spaziano dalla visibilità di Akamai sull'attività dei bot a livello mondiale alle firme di bot personalizzate e al rilevamento in tempo reale dei bot sconosciuti. L'intervento del Bot Manager è pertanto automatico e colpisce i singoli bot in base all'attività di web scraping rilevata in precedenza fra tutti i clienti Akamai affidandosi al traffico proveniente da bot sconosciuti tramite analisi comportamentale, browser fingerprint, rilevamento automatizzato del browser rilevamento delle anomalie http e dal tasso di richieste elevato. Un'altra funzionalità offerta dal Bot Manager è quella di consentire alle attività che decidono di utilizzare tale soluzione di creare firme e categorie di bot personalizzate al fine di identificare bot specifici che interagiscono regolarmente con il proprio sito web e di assegnare diverse azioni da intraprendere che comprendono bloccare, ritardare, fornire contenuti alternativi, captcha o PoW (Proof of Work).

Per i motivi sopra elencati realizzare un software per automatizzare il processo di acquisto su piattaforme che utilizzano tale tipologia di protezione risulta estremamente complesso. Tuttavia, è possibile, tramite richieste all'API Akamai costruire dei cookie che, se utilizzati mentre si effettuano richieste, come ad esempio l'aggiunta al carrello di un articolo, possono mascherare l'attività dei bot fingendosi attività umana. Esempio di un cookie valido Akamai è il seguente:

```
"3916BBB63A3F6462E9CDF4BA60F8CAC3~1~YAAQjZgRYGeITbxxAQAAYa2KwQPe5  
pjrnR/uFkswmD/l3F2+Yu+AXA5CytKRomfM6HHatpf6crAZBWSUXfhMXUeMI8X9qt8W77  
Sm9qn1rE51e8uEBDxpbHfY9d3TzeJ+dVMp+LevXLzG5EfABet2FfaSaoF+3V2eu1cDFS0j  
HBwpSP1aNMGEmZSziWLW3z1G5nrGb8WZ3cfH4N2Kcl5o4k9ryztuW4iA6jk/Y603NFq67  
Sd8hLeT3h8DRql2bewGW3DfekwhG+3BflMf0PkDmXxDAebhSGXjcSO7wqf5wGYEFVfc  
wulCpx4qm+VxwAy+AKerQ0MlzGX0LGPxfJZ62tFpBAw=~-1~-1~-1"
```

2.2.4 Datadome

Datadome rappresenta un'ottima alternativa al più costoso Akamai per quanto riguarda un servizio di protezione real-time dai bot. A differenza dei concorrenti, esso propone una soluzione ibrida per identificare se il visitatore sia umano o meno. Tale soluzione si basa sulla più comune forma di rivelazione lato server che sfrutta tecniche di fingerprint HTTP, TCP, TLS e di analisi del comportamento che riguarda lo studio del numero di richieste provenienti dallo stesso indirizzo IP, la loro frequenza o comunque il presentarsi di uno specifico pattern ritenuto non umano, a cui viene aggiunta una rivelazione lato client o "in-browser". Questo porta a un'analisi più dettagliata su un ampio panorama di fattori minori come ad esempio sfide JavaScript, risoluzione dello schermo, numero di tocchi in un punto, numero di processori attivi, movimenti del mouse e azioni dovuti a click. Come sostiene la casa proprietaria di tale soluzione:

“Questi segnali lato client sono cruciali al fine di rilevare i bot più avanzati, soprattutto quando quest’ultimi utilizzano una specifica fingerprint al fine di eludere sistemi di sicurezza meno sofisticati.” [37]

2.2.5 Variti

Variti rappresenta la soluzione meno utilizzata in termini di scelta come infrastruttura di protezione dai bot. Questa scarsa richiesta è dovuta solo alla poca sponsorizzazione del proprio prodotto e non dalla sua efficacia. Variti Bot Protection a differenza dei suoi concorrenti, offre un accurato sistema di protezione dell’API proprietario del sito. In questo modo lo scraping del sito risulta estremamente difficile agli sviluppatori di bot. L’operazione di scraping è infatti il punto cardine per la progettazione di un adeguato algoritmo per l’acquisto di prodotti. Tramite lo scraping del sito è infatti possibile trovare i prodotti caricati, individuare l’indirizzo dell’articolo che si desidera acquistare e infine verificare quali taglie sono disponibili al momento, oltre che visualizzare le corrette richieste. Tale forma di protezione è utilizzata dal colosso dei brand streetwear di lusso Off-White. A tal proposito, viene riportato di seguito il resoconto del rilascio sul sito Off-White protetto dal sistema Variti delle Jordan 4 Off White, una delle sneaker più richieste dell’anno. L’articolo scritto da Riccucci Pietro [38], noto sviluppatore di software legati al mondo delle sneaker, riporta come i responsabili del reparto Web di Off-White, abbiano reso peggiore l’esperienza degli utenti manuali bloccando a chiunque l’accesso al /api/ endpoint con un 302 che ridirigeva alla pagina principale se “ss” e “_cfuid” non erano presenti nella sessione. L’autore sottolinea inoltre come gli sviluppatori abbiano facilmente intuito come eludere questi controlli semplicemente aggiungendo i seguenti valori alla sessione:

```
ss=a; _cfuid=a
```

Fig. 2.14 Vincolo per accedere alla pagina Off White durante il rilascio del prodotto in edizione limitata

Gli sviluppatori del negozio online, al fine di bloccare i bot, decisero di aggiungere una convalida tramite SMS, la quale costringeva gli utenti manuali ad inserire il proprio numero di telefono prima di aggiungere al carrello il prodotto. Nell'articolo viene ribadito come questa non fosse una cattiva idea, ma che l'implementazione avrebbe funzionato nel 2015.



Fig. 2.15 Pagina relativa alle Nike Air Jordan 4 Retro sul sito Off-White

```
POST
https://www.off---white.com/api/marketing/v1/verificationcodes

{channel: "SMS",
recipient: "+393914568912"}
```

Fig. 2.16 Richiesta POST necessaria per ricevere il codice di verifica

Una volta ricevuto il messaggio, l'utente avrebbe dovuto confermare il token ricevuto effettuando una nuova richiesta e confermando la propria identità. Sfortunatamente, la convalida tramite SMS non era necessaria per aggiungere il prodotto al carrello, in quanto bastava effettuare una semplice richiesta POST all'endpoint del carrello senza

dover convalidare la propria identità. Per questi motivi, nessun utente umano riuscì ad acquistare le tanto ambite Air Jordan 4 in collaborazione con Off White.

2.2.6 PerimeterX

PerimeterX è un fornitore di sistemi di sicurezza informatica ad alta scalabilità basati sull'analisi del comportamento per il web, il cloud e le piattaforme mobili. Analizzando il comportamento degli utenti umani, delle applicazioni e dei bot, PerimeterX è in grado di individuare in real-time attacchi da parte di bot con estrema accuratezza. L'intera gestione della piattaforma di sicurezza è affidata a PerimeterX Bot Defender, il quale, come suggerisce il nome, è un bot specializzato nella protezione e nella salvaguardia di siti web, applicazioni mobili e più in generale di API.

Please verify you are a human

TENERE PREMUTO

Access to this page has been denied because we believe you are using automation tools to browse the website.

This may happen as a result of the following:

- Javascript is disabled or blocked by an extension (ad blockers for example)
- Your browser does not support cookies

Please make sure that Javascript and cookies are enabled on your browser and that you are not blocking them from loading.

Reference ID: 5c973910-e87a-11ea-b049-99e4438e385c

Fig. 2.17 Tentativo di accesso al sito Solebox durante il rilascio di un prodotto in edizione limitata.

Esso sfrutta l'apprendimento autonomo per aggiornare costantemente la propria libreria di pattern di comportamenti sospetti basandosi su miliardi di visitatori su siti web, siti mobile e applicazioni web. Esempi di e-commerce che utilizzano tale soluzione sono Snipes, Solebox e Onygo.

2.3 Caso d'interesse: Shopify

La scelta di analizzare nello specifico Shopify non risiede esclusivamente nel suo enorme utilizzo in termini di piattaforma, ma in particolare dall'API di quest'ultima. Infatti le richieste necessarie, ad esempio, per aggiungere al carrello o completare l'ordine sono le stesse per tutti gli shop online basati su Shopify. Questo non si verifica nelle restanti piattaforme precedentemente analizzate, le quali offrono la possibilità di personalizzare le diverse richieste per completare un ordine. In questo caso analizzerò il sito web di uno dei più importanti negozi italiani in termini di streetwear che ha recentemente cambiato la propria piattaforma e-commerce adattandola a Shopify: One Block Down disponibile all'indirizzo <https://it.oneblockdown.it/>.

2.3.1 I prodotti

Come ogni negozio fisico mantiene nei propri registri l'inventario, anche un sito e-commerce necessita il mantenimento delle informazioni relative ai prodotti attualmente disponibili. Nel caso di Shopify, il catalogo dei prodotti è facilmente consultabile, visitando l'URL del dominio del negozio con path “/products.json”. Nel nostro caso “<https://it.oneblockdown.it/products.json>”.

```
{
  "name": "Colore",
  "position": 2,
  "values": [
    "Multicolore"
  ]
}, {
  "id": "4456938471476",
  "title": "Andy Warhol Drew Flowers Rosso",
  "handle": "andy-warhol-drew-flowers-rosso",
  "body_html": "Parte di una serie di occhiali da sole SUPER-ANDY WARHOL, questo modello interpreta lo stile distintivo del grande artista pop con un design tridimensionale ricco e stratificato. Una silhouette Drew Feminine è stata abbinata ad elaborati decori floreali sia sulla montatura frontale che sulle aste, risultando in un design giocoso ma comodo e facile da abbinare. <br>",
  "published_at": "2020-08-07T19:19:38+02:00",
  "created_at": "2020-01-25T00:58:18+01:00",
  "updated_at": "2020-08-07T19:19:38+02:00",
  "vendor": "Retrosuperfuture",
  "product_type": "Occhiali",
  "tags": [
    "#3F8",
    "IM3C0012",
    "2020-01-24T23:58:18Z",
    "color_Rosso",
    "filter-brand_Retrosuperfuture",
    "filter-category_Occhiali",
    "filter-clothingsize_OS",
    "filter-color_Rosso",
    "filter-macro_Accessori",
    "filter-macro_accessories",
    "filter-price_€200 / €400",
    "filter-type_Occhiali",
    "gender_unisex",
    "last_piece",
    "photo_yes",
    "republished",
    "season_S52019",
    "sizegrid_OS",
    "SpecialSale",
    "standard",
    "style_IM3C0012"
  ],
  "variants": [
    {
      "id": "31633245962292",
      "title": "OS / Rosso",
      "option1": "OS",
      "option2": "Rosso",
      "option3": null,
      "sku": "IM3C000C0012R05",
      "requires_shipping": true,
      "taxable": true,
      "featured_image": {
        "id": "14531466199992",
        "product_id": "4456938471476",
        "position": 1,
        "created_at": "2020-06-24T10:34:12+02:00",
        "updated_at": "2020-06-24T10:34:12+02:00",
        "alt": null,
        "width": 1333,
        "height": 2000,
        "src": "https://cdn.shopify.com/s/files/1/0070/1484/2420/products/IM3C0012_1.png?v=1592987652",
        "variant_ids": [
          "31633245962292"
        ]
      },
      "available": true,
      "price": "300.00",
      "grams": 300,
      "compare_at_price": "0.00",
      "position": 1,
      "product_id": "4456938471476",
      "created_at": "2020-01-25T00:58:18+01:00",
      "updated_at": "2020-08-07T18:03:46+02:00",
      "images": [
        {
          "id": "14531466199992",
          "created_at": "2020-06-24T10:34:12+02:00",
          "position": 1,
          "updated_at": "2020-06-24T10:34:12+02:00",
          "product_id": "4456938471476",
          "variant_ids": [
            "31633245962292"
          ]
        },
        {
          "src": "https://cdn.shopify.com/s/files/1/0070/1484/2420/products/IM3C0012_1.png?v=1592987652",
          "width": 1333,
          "height": 2000,
          "id": "14531466231860",
          "created_at": "2020-06-24T10:34:12+02:00",
          "position": 2,
          "updated_at": "2020-06-24T10:34:12+02:00",
          "product_id": "4456938471476",
          "variant_ids": [
            "31633245962292"
          ]
        }
      ],
      "options": [
        {
          "name": "Taglia",
          "position": 1,
          "values": [
            "OS"
          ]
        },
        {
          "name": "Colore",
          "position": 2,
          "values": [
            "Rosso"
          ]
        }
      ]
    }, {
      "id": "4646310608948",
      "title": "6 Retro Quai 54",
      "handle": "6-retro-quai-54",
      "body_html": "La storia del brand Jordan è legata a doppio filo a Parigi. Questa A16 rende omaggio ai campi in asfalto del torneo di streetball Quai 54, sponsorizzato da Jordan, che si tiene annualmente nella Ville Lumière. La pelle di alta qualità e i materiali in tessuto di questa A16 presentano colori che si abbinano alla perfezione con la collezione di abbigliamento Quai 54. <!-- split --><iframe src='\"https://docs.google.com/forms/d/e/1FAIpQLSdyobg10m8B9mI2_s5uAt_V0vXckj80W0Fv2PMs0nHg/viewform?embedded=true\"' width='640' height='1361' frameborder='0' marginheight='0' marginwidth='0'>Caricamento...</iframe>",
      "published_at": "2020-08-07T12:08:06+02:00",
      "created_at": "2020-07-27T17:42:14+02:00",
      "updated_at": "2020-08-07T12:10:09+02:00",
      "vendor": "Jordan Brand",
      "product_type": "Sneakers",
      "tags": [
        "#3F8",
        "carryover",
        "color_Beige",
        "countdown",
        "countdown-finish_14/08/2020 09:00:00",
        "CZ4152-101",
        "drops",
        "filter-brand_Jordan Brand",
        "filter-category_Sneakers",
        "filter-color_Beige",
        "filter-macro_Calzature",
        "filter-macro_footwear",
        "filter-price_€100 / €200",
        "filter-type_Sneakers",
        "gender_unisex",
        "hot",
        "jordan6",
        "limited_1",
        "new",
        "no-stock",
        "photo_yes",
        "raffle_open",
        "season_FW2020",
        "sizegrid_SM2",
        "style_CZ4152-101"
      ],
      "variants": [
        {
          "id": "32234500259899",
          "title": "8 / Beige",
          "option1": "8",
          "option2": "Beige",
          "option3": null,
          "sku": "CZ4152-#0000#101#8",
          "requires_shipping": true,
          "taxable": true,
          "featured_image": {
            "id": "14593770160180",
            "product_id": "4646310608948",
            "position": 1,
            "created_at": "2020-07-27T17:42:17+02:00",
            "updated_at": "2020-07-27T17:42:23+02:00",
            "alt": null,
            "width": 1333,
            "height": 2000,
            "src": "https://cdn.shopify.com/s/files/1/0070/1484/2420/products/CZ4152-101_1_4c48c31a-de07-440f-a3f9-fc03d07945a.png?v=1595864543",
            "variant_ids": [
              "32234500259899",
              "32234500292660",
              "32234500325428",
              "32234500358196",
              "32234500390964",
              "32234500423732",
              "32234500456500",
              "32234500489268",
              "32234500522036",
              "32234500554804",
              "32234500587572"
            ]
          },
          "available": false,
          "price": "190.00",
          "grams": 1200,
          "compare_at_price": "0.00",
          "position": 1,
          "product_id": "4646310608948",
          "created_at": "2020-07-27T17:42:15+02:00",
          "updated_at": "2020-07-27T17:42:23+02:00",
          "id": "32234500292660",
          "title": "8.5 / Beige",
          "option1": "8.5",
          "option2": "Beige",
          "option3": null,
          "sku": "CZ4152-#0000#101#8.5",
          "requires_shipping": true,
          "taxable": true,
          "featured_image": {
            "id": "14593770160180",
            "product_id": "4646310608948",
            "position": 1,
            "created_at": "2020-07-27T17:42:17+02:00",
            "updated_at": "2020-07-27T17:42:23+02:00"
          }
        }
      ]
    }
  ]
}
```

Fig. 2.18 Pagina restituita all'indirizzo <https://it.oneblockdown.it/products.json>

Come viene mostrato dall'immagine, l'url riporta ad una pagina in formato Json che contiene tutte le informazioni relative ai prodotti presenti all'interno del sito web, come il nome del prodotto, utilizzato per ricercare il prodotto interessato tramite parole chiave, il variant che identifica univocamente il prodotto, l'url del prodotto stesso e infine tutte le taglie presenti con i relativi quantitativi.

Poiché spesso non tutti i prodotti vengono caricati nel file in formato Json, risulta utile verificare la loro presenza anche all'indirizzo formato sempre dall'URL del sito Shopify d'interesse seguito da `"/collections/all/products.atom"` il quale restituisce una pagina contenente tutti i prodotti disponibili descritti in formato Atom, un formato di documento basato su XML per la sottoscrizione di contenuti web.

2.3.2 Aggiunta al carrello

Una volta reperiti tutti i prodotti e scelto quello di interesse mediante parole chiavi è necessario aggiungere al carrello la taglia desiderata. Tale operazione è molto semplice, basterà infatti effettuare una richiesta GET all'indirizzo formato dall'indirizzo del sito seguito da `"/cart/add.js?quantity=1&id=sizeVariant"` dove `sizeVariant` rappresenta l'identificativo univoco della taglia del prodotto selezionato, prelevabile dal file Json o Atom precedentemente elaborato.

Nel caso di `https://it.oneblockdown.it/collections/calzature/products/overreact-sandal-ispaspa-nero` trovato utilizzando le parole chiavi `"overreact"` e `"ispa"`, il `sizeVariant` relativo alla taglia 8 americana è `32222754177076`. Di conseguenza l'URL a cui effettuare la richiesta risulterà:

`https://it.oneblockdown.it/cart/add.js?quantity=1&id=32222754177076`.

Tale richiesta, riceverà una risposta in formato json contenente le informazioni relative al prodotto aggiunto al carrello. Inoltre, controllando se la quantità presente nel Json di risposta è pari a quella indicata come parametro al link precedente, è possibile

verificare se l'oggetto è stato correttamente aggiunto al carrello oppure si sono verificati problemi durante l'esecuzione della richiesta.

```
{
  "id": "32222754177076",
  "properties": null,
  "quantity": 1,
  "variant_id": "32222754177076",
  "key": "32222754177076:7e14bfe20a4e2c21e0074d6fb63b7993",
  "title": "Overreact Sandal ISPA Nero - 8 \\ Nero",
  "price": 16000,
  "original_price": 16000,
  "discounted_price": 16000,
  "line_price": 16000,
  "original_line_price": 16000,
  "total_discount": 0,
  "discounts": [],
  "sku": "CQ2230-0000001#8",
  "grams": 1200,
  "vendor": "Nike",
  "taxable": true,
  "product_id": 4642937733172,
  "product_has_only_default_variant": false,
  "gift_card": false,
  "final_price": 16000,
  "final_line_price": 16000,
  "url": "\/products\/overreact-sandal-isp-nero?variant=32222754177076",
  "featured_image": {
    "aspect_ratio": 0.667,
    "alt": "Overreact Sandal ISPA Nero",
    "height": 2000,
    "url": "https:\/\/cdn.shopify.com\/s\/files\/1\/0070\/1484\/2420\/products\/CQ2230-001_1.png?v=1595244087",
    "width": 1333
  },
  "image": "https:\/\/cdn.shopify.com\/s\/files\/1\/0070\/1484\/2420\/products\/CQ2230-001_1.png?v=1595244087",
  "handle": "overreact-sandal-isp-nero",
  "requires_shipping": true,
  "product_type": "Sneakers",
  "product_title": "Overreact Sandal ISPA Nero",
  "product_description": "Il sandalo Nike ISPA OverReact offre un comfort ideale grazie a un'intersuola incredibilmente ammortizzata e a una tomaia leggera, ideale quando fa caldo. L",
  "variant_title": "8 \\ Nero",
  "variant_options": [
    {
      "name": "Taglia",
      "value": "8"
    },
    {
      "name": "Colore",
      "value": "Nero"
    }
  ],
  "line_level_discount_allocations": [],
  "line_level_total_discount": 0
}
```

Fig. 2.19 Body della response in seguito alla richiesta <https://it.oneblockdown.it/cart/add.js?quantity=1&id=32222754177076>

2.3.3 Checkout

Una volta aggiunto il prodotto al carrello con successo è possibile inizializzare il processo di checkout. Al fine di effettuare correttamente il checkout è necessario utilizzare un token di autenticità utilizzato dalla piattaforma Shopify per verificare l'identità dell'utente. Il token viene ricavato dal file Json salvato all'indirizzo formato dal sito seguito da “/checkout.json”. Una volta ottenuto il token di autenticità è possibile passare allo step successivo. A tal punto, in caso di forte traffico sul sito, è possibile incomberne nella coda Shopify, ovvero un sistema che mette in coda le richieste in modo da riuscire a rispondere a tutte senza causare ritardi o crash del sistema. Il passaggio della coda non sembra essere effettuato in modalità “first come first served”, in cui le richieste effettuate per prima sono anche quelle che passano la coda più in fretta, ma del tutto casuale. Sarà necessario dunque verificare periodicamente se attualmente ci si trova in stato di coda o meno semplicemente controllando a intervalli di tempo regolari se nell'attuale URL è presente la parola chiave “queue” (coda in inglese). Questa modalità di rilascio casuale della possibilità di acquistare mostra quanto siano necessari l'uso di bot per acquistare i prodotti desiderati. Mentre un utente manuale potrà effettuare una sola richiesta per volta, affidandosi esclusivamente alla propria fortuna per passare la coda il prima possibile, i bot effettueranno centinaia di richieste nello stesso momento aumentando

notevolmente le probabilità di successo, e di conseguenza di acquisto del prodotto desiderato.

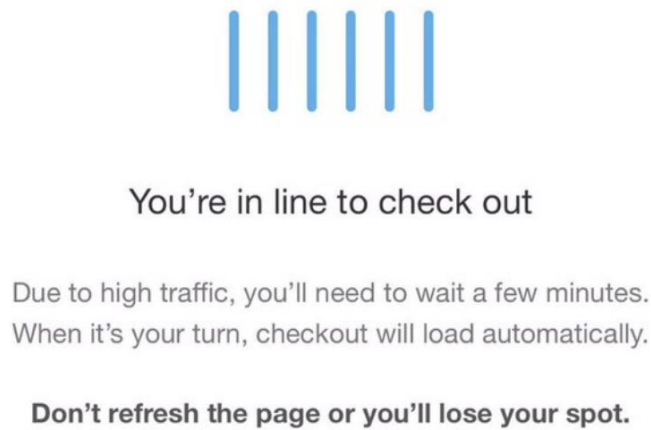


Fig. 2.20 Pagina di coda caratteristica di Shopify

Superata la coda, è possibile effettuare una richiesta POST alla pagina di checkout. Tale richiesta contiene nel proprio body tutte le informazioni relative ai dati di spedizione dell'utente.

```

Form Data  view source  view URL encoded
_method: patch
authenticity_token: XdP+2Q1r-i0B2G4pPo+MevghszHYcDCelwzTuVZghxCrYmSScLmtS37ZEz1VJnffMs3zMo31R3nM+1GpG5QMgnGFQ==
previous_step: contact_information
step: shipping_method
checkout[email]:
checkout[shipping_address][first_name]: Giuseppe
checkout[shipping_address][last_name]: Cristaudo
checkout[shipping_address][company]:
checkout[shipping_address][address1]:
checkout[shipping_address][address2]:
checkout[shipping_address][city]:
checkout[shipping_address][country]:
checkout[shipping_address][province]:
checkout[shipping_address][zip]:
checkout[shipping_address][phone]:
checkout[shipping_address][first_name]: Giuseppe
checkout[shipping_address][last_name]: Cristaudo
checkout[shipping_address][company]:
checkout[shipping_address][address1]:
checkout[shipping_address][address2]:
checkout[shipping_address][city]:
checkout[shipping_address][country]:
checkout[shipping_address][province]:
checkout[shipping_address][zip]:
checkout[shipping_address][phone]:
checkout[remember_me]:
checkout[remember_me]: 0
checkout[client_details][browser_width]: 2543
checkout[client_details][browser_height]: 1297
checkout[client_details][javascript_enabled]: 1
checkout[client_details][color_depth]: 24
checkout[client_details][java_enabled]: false
checkout[client_details][browser_tz]: -120

```

Fig. 2.21 Body richiesta POST contenente le informazioni di spedizione

Come si può osservare, oltre ai dati di spedizione utente, sono presenti il token di autenticità e alcune informazioni relative al browser utilizzato.

Successivamente è possibile ricavare i costi relativi alla spedizione. Tale operazione non è necessaria ma può essere utile per verificare se il negozio dispone di spedizione nel proprio stato, oppure per selezionare una spedizione espressa piuttosto che una standard. I valori sono facilmente ricavabili dalla risposta in formato json che si ottiene effettuando una richiesta GET all'indirizzo del sito web seguito da

“/cart/shipping_rates.json?shipping_address[zip]=zipCode&shipping_address[country]=country&shipping_address[province]=province”

Dove zipCode, country e province rappresentano rispettivamente il codice postale, lo stato e la provincia relativa al proprio indirizzo di spedizione. Una volta ottenuti gli identificativi della spedizione è possibile effettuare nuovamente una richiesta di POST al cui interno sono contenute le informazioni relative alla spedizione utilizzata.

```

_method: patch
authenticity_token: T65LFR38yCuRKmn8nsHN6goxDSYI/e9p54rFSuYJovg0NJKvikPLZnYCauFaWJhdzzf22QaW+xZsFTN80bFuMw==
previous_step: shipping_method
step: payment_method
checkout[shipping_rate][id]: shopify-Spedizione%20Gratuita-0.00
checkout[client_details][browser_width]: 1872
checkout[client_details][browser_height]: 1297
checkout[client_details][javascript_enabled]: 1
checkout[client_details][color_depth]: 24
checkout[client_details][java_enabled]: false
checkout[client_details][browser_tz]: -120

```

Fig. 2.22 Body richiesta POST contenente i metodi disponibili per la spedizione

Impostata la spedizione è ora possibile effettuare il pagamento. Esso come nei casi precedenti viene impostato effettuando una richiesta POST all'indirizzo "https://deposit.us.shopifycs.com/sessions", contenente nel payload tutti i dati relativi al metodo di pagamento utilizzato. In tal caso si è scelto di sviluppare solo pagamenti tramite carta di credito.

```

▼ Request Payload    view source
  ▼ {credit_card: {number: , name: , , , ,}}
    ▼ credit_card: {number: , name: , , , ,}
      month:
      name:
      number:
      verification_value:
      year:

```

Fig. 2.23 Body richiesta POST contenente le informazioni di pagamento

Infine, si può completare l'ordine effettuando l'ultima richiesta POST all'url relativo al proprio acquisto ottenuto in precedenza. Nuovamente viene utilizzato il medesimo token di autenticità per garantire l'identità dell'utente nel passaggio finale.

```
▼ Form Data    view source    view URL encoded
_method: patch
authenticity_token: +hP2Tn42fLbAxw91VeI3/y1iX8CHktS8UynVsnmaKRSBiS+c6Yl+GyfVdGiRe2JI6GSkP4n5wMPYtiOETiL13w==
checkout[total_price]: 16000
complete: 1
checkout[client_details][browser_width]: 1261
checkout[client_details][browser_height]: 1297
checkout[client_details][javascript_enabled]: 1
checkout[client_details][color_depth]: 24
checkout[client_details][java_enabled]: false
checkout[client_details][browser_tz]: -120
```

Fig. 2.21 Body richiesta POST utilizzata per completare l'ordine

2.3.4 Sistemi per fermare i bot

Nonostante Shopify risulti la piattaforma maggiormente utilizzata dagli e-commerce, i meccanismi che mette a disposizione per facilitare l'acquisto manuale sono considerabili indifferenti. Infatti, nessun sito web che si basa su Shopify utilizza infrastrutture di sicurezza all'Avanguardia come Akamai o Datadome, ma si affida esclusivamente agli strumenti offerti direttamente dalla piattaforma. Tale semplicità rende l'acquisto manuale praticamente impossibile, confermando la necessità di adattarsi a queste nuove modalità di acquisto. Esistono principalmente due soluzioni per rallentare i bot:

- Google reCaptcha V2, il quale come trattato in precedenza non risulta essere un'ottima soluzione per bloccare i bot vista l'ampia gamma di metodologie per eluderlo. I Captcha sono così inseriti in alcuni punti chiave dell'acquisto che possono essere il login, l'aggiunta al carrello o il completamento dell'ordine. Nel caso di Shopify la soluzione utilizzata dalla maggior parte dei bot è il "manual captcha harvester", il quale permette di risolvere manualmente la sfida offerta dal Captcha. L'utilizzo di tale soluzione deriva dalla velocità dell'operazione, infatti l'utilizzo di captcha farms risulterebbe decisamente più lento rispetto ad una soluzione manuale.

- Proxy throttling, essenzialmente il proprietario del sito web è a conoscenza di un range di indirizzi di proxy utilizzati da chi acquista utilizzando bot e fa in modo che chi acceda al proprio server con questi range di indirizzi non sia in grado, ad esempio, di trovare i prodotti oppure di completare l'ordine. Un proxy non è altro che un server che funge da intermediario per le richieste da parte dei clienti alla ricerca di risorse su altri server, disaccoppiando l'accesso al web dal browser. La soluzione a tale tipologia di mitigazione dei bot è semplice e consiste nel lanciare meno richieste utilizzando il proprio indirizzo IP.

Infine, negli ultimi mesi, Shopify ha sviluppato una sfida proprietaria che consiste nell'effettuare una domanda relativa al prodotto nel momento dell'aggiunta del carrello. Tuttavia, anche questa soluzione risulta essere inefficace poiché basta che il bot sia in grado di effettuare uno scraping della pagina al fine ricavare la domanda e successivamente eseguire una richiesta POST contenente la risposta.

3. Specifiche progettuali

Il seguente capitolo tratta dell'intero processo di sviluppo del progetto, specificando oltre alle richieste che vengono effettuate ai siti anche il funzionamento e la configurazione che permettono il corretto funzionamento dell'intero sistema.

L'utente si interfaccia con l'applicazione attraverso una semplice e immediata interfaccia CLI (command line interface). L'interfaccia a riga di comando è caratterizzata da un'interazione testuale tra utente e applicazione dove l'utente impartisce comandi testuali in input mediante tastiera e riceve risposte testuali sullo standard output. Questo permette l'utilizzatore di avviare il sito desiderato nella maniera più immediata e veloce possibile, caratteristiche necessarie per l'ambito di utilizzo. Nonostante la tendenza attuale degli sneaker bot vada verso l'utilizzo di interfacce grafiche o GUI, si è ritenuto che l'applicazione si dovesse concentrare più sull'efficienza dell'algoritmo per acquistare prodotti piuttosto che rendere più accattivante l'interfaccia.

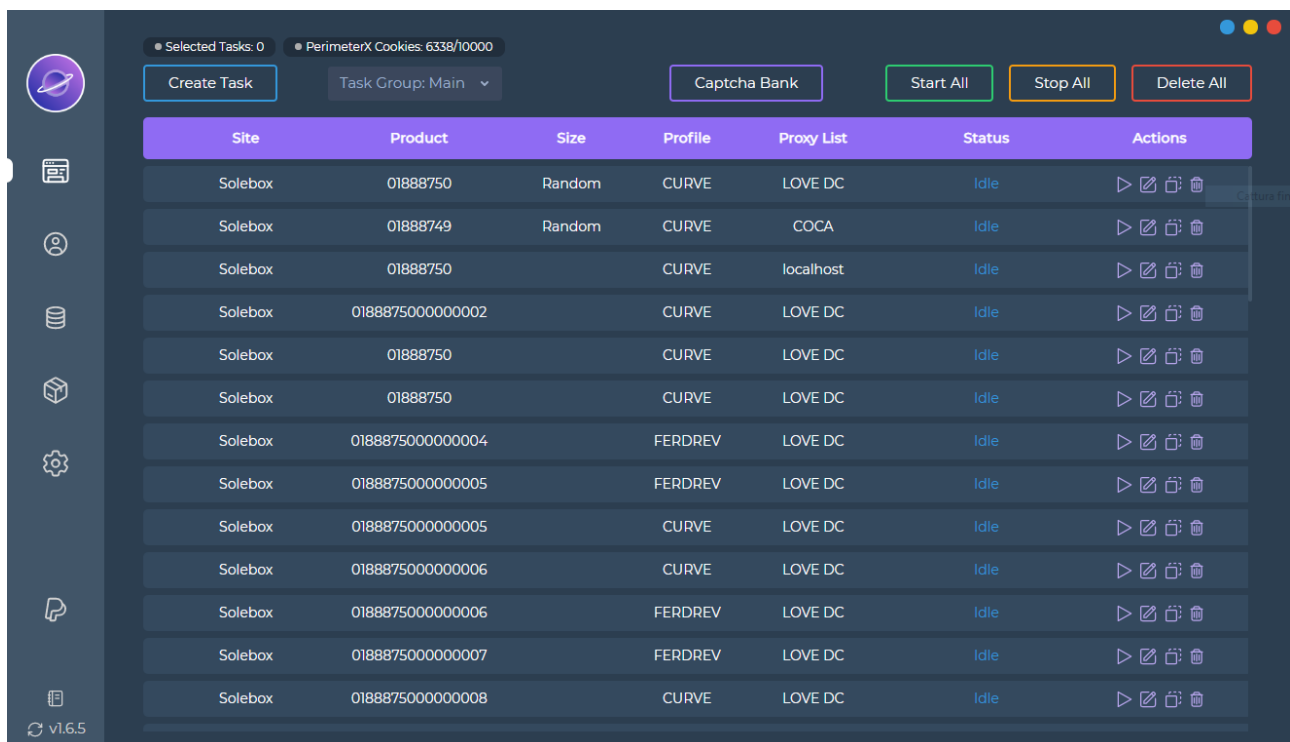


Fig. 3.1 Interfaccia grafica (GUI) di Orbit, attualmente ritenuto uno dei migliori sneaker bot europei

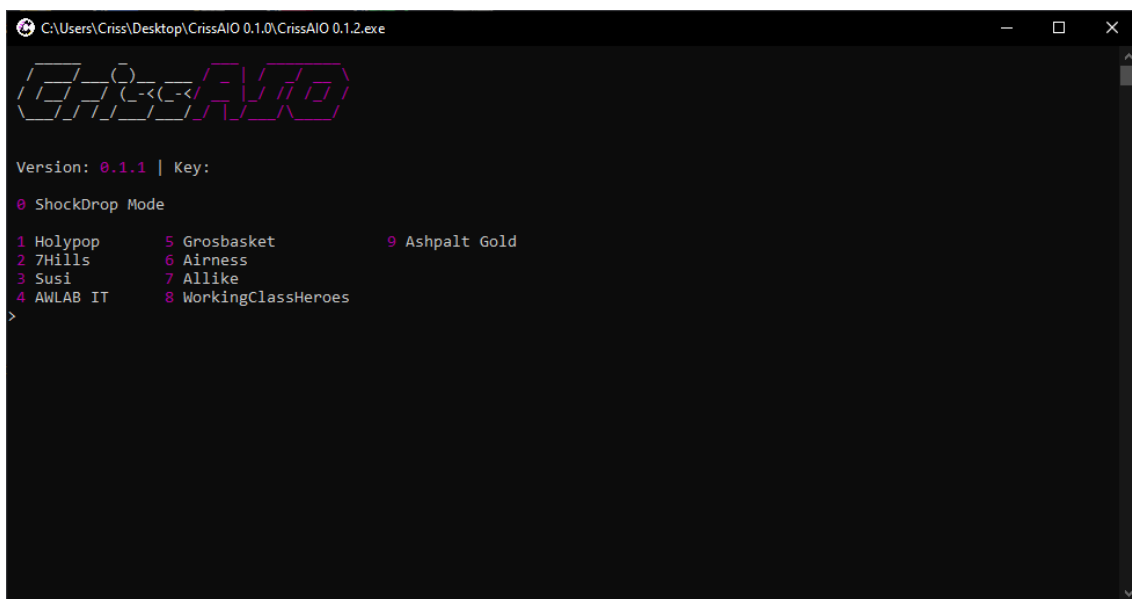


Fig. 3.2 Interfaccia grafica (CLI) dello sneaker bot sviluppato

Nella schermata sono presenti tutti i siti attualmente supportati, selezionabili immettendo nella riga di input il numero corrispondente e premendo invio. Ogni sito utilizza una cartella di supporto da cui prelevare le informazioni necessarie a completare l'acquisto sulla piattaforma desiderata.

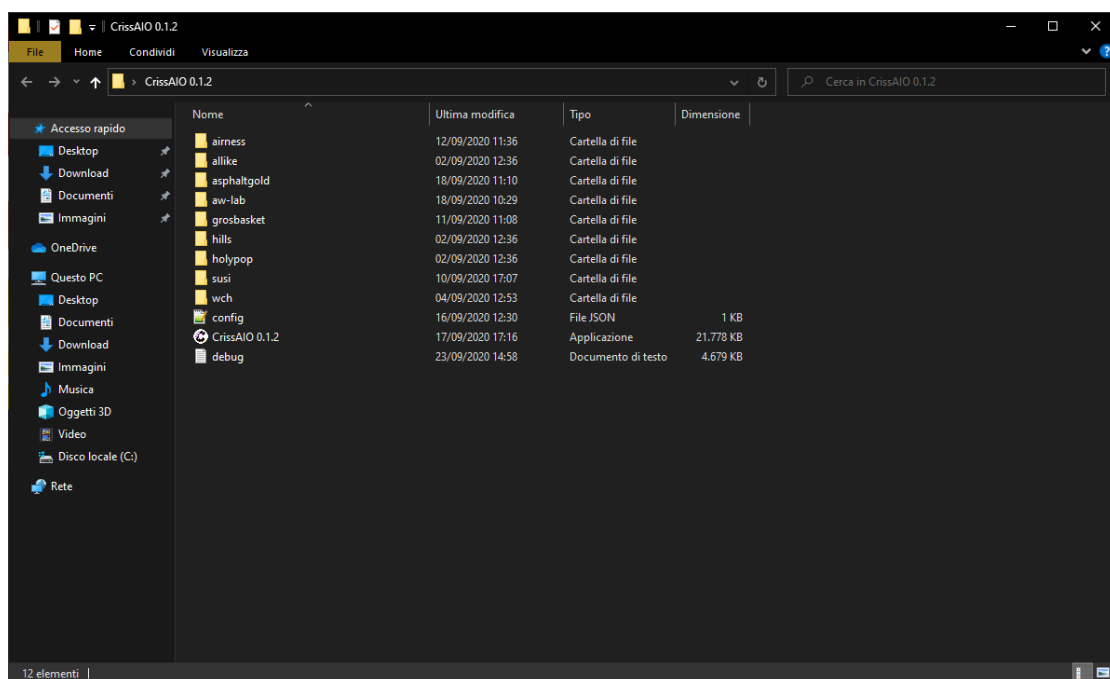
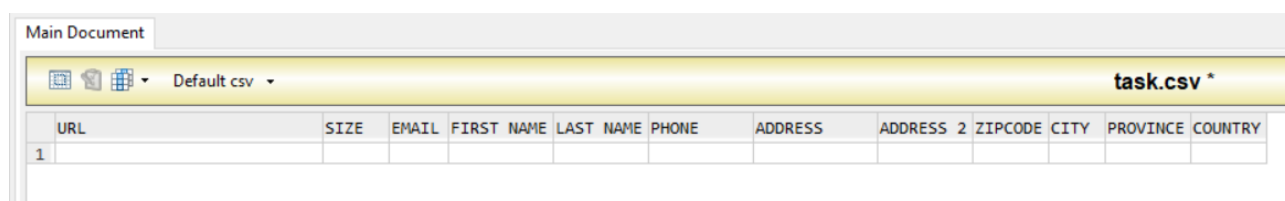


Fig. 3.3 Cartella contenente tutti i file dell'applicazione sviluppata

All'interno di ogni cartella sono dunque previsti due file: task.csv e proxies.txt.

Il formato csv, o comma-separated values, è un formato di file in cui ogni riga della tabella è normalmente rappresentata da una linea di testo divisa in campi, da un apposito carattere, ciascuno dei quali rappresenta un valore.

Il formato csv è stato scelto appunto per la gestione delle task per la sua facilità di importazione ed esportazione della tabella dei dati. Ogni sito prevede una propria formattazione, tuttavia tutti contengono principalmente l'URL del prodotto interessato, la taglia desiderata e una serie di informazioni personali per il completamento dell'ordine. Inoltre, ogni riga rappresenta un thread che andrà ad effettuare l'acquisto del prodotto in maniera indipendente dalle altre task.



	URL	SIZE	EMAIL	FIRST NAME	LAST NAME	PHONE	ADDRESS	ADDRESS 2	ZIPCODE	CITY	PROVINCE	COUNTRY
1												

Fig. 3.4 Esempio file CSV supportato (per migliorare l'effettiva visualizzazione del formato csv è stato utilizzato il software Rons editor [])

In secondo luogo, il file proxies.txt verrà accuratamente scritto dall'utente, il quale inserirà una lista dei proxy che desidera utilizzare. Un proxy è un server utilizzato per lo scambio di richieste tra un cliente ed un server, fungendo da intermediario.

In questo modo, in caso una determinata richiesta venisse bannata temporaneamente dal sito sarà possibile effettuarla nuovamente semplicemente cambiando proxy e fingendosi per il server del sito un altro utente. La funzione utilizzata per salvare i proxy è la seguente:

```

def loadProxies():

    with open('airness/proxies.txt', 'r') as proxies_file:
        file_raw = proxies_file.read().split('\n')

    proxy_raw = random.choice(file_raw)

    try:
        proxy_parts = proxy_raw.split(':')
        ip, port, user, password = proxy_parts[0], proxy_parts[1], proxy_parts[2], proxy_parts[3]
        new_proxy = {
            'http': f'http://{user}:{password}@{ip}:{port}',
            'https': f'https://{user}:{password}@{ip}:{port}'
        }
        global proxy
        proxy = new_proxy
    except Exception as e:
        print(threadTime() + bcolors.WARNING + 'No proxies found, running localhost..' + bcolors.ENDC)
        logging.info('[AIRNESS] ' + utility.threadTime() + 'No proxies found, running localhost..')
        pass

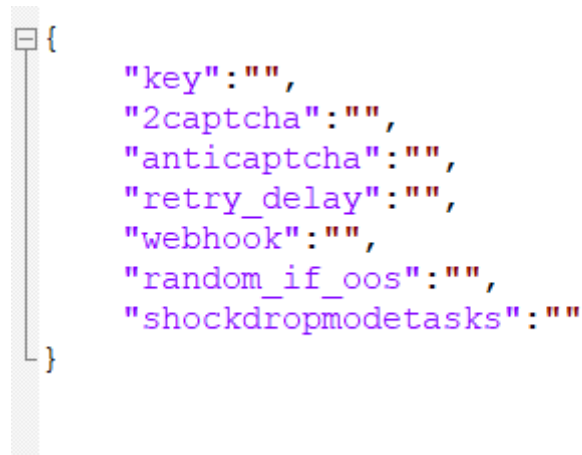
```

Fig. 3.4 Algoritmo utilizzato per creare oggetti proxy a partire dal file proxies.txt

Il funzionamento si basa sul prelevare una riga casuale del file, suddividere un proxy in username, password, IP e porta e di creare un oggetto proxy da utilizzare successivamente durante le richieste. Nel caso in cui non fosse presente alcun proxy all'interno del file, il bot procederà a eseguire le richieste utilizzando l'indirizzo IP della macchina su cui è stato lanciato.

La coppia di file trattati risulta essere la soluzione più efficiente per costruire la base di dati dell'algoritmo. Nonostante la creazione di task sembri macchinosa, questa si riduce a poche semplici operazioni, facilitate dagli strumenti messi a disposizione dai software di editor dei file csv. Questa tesi viene confermata dal fatto che tutti gli sneaker bot basati su interfaccia CLI attualmente in circolazione utilizzino questa tipologia di struttura.

Infine, la configurazione del sistema può essere effettuata modificando i parametri contenuti nel file “config.json”.



```
{
  "key": "",
  "2captcha": "",
  "anticaptcha": "",
  "retry_delay": "",
  "webhook": "",
  "random_if_oos": "",
  "shockdropmodetasks": ""
}
```

Fig. 3.5 File config.json contenuto all'interno della cartella dell'applicazione

La scelta del formato json ricade soprattutto per la sua facilità di consultazione. Dal file si possono quindi inserire la propria chiave di autenticazione, il cui sistema di funzionamento verrà specificato nel paragrafo successivo, due parametri “anticaptcha” e “captcha” in cui è possibile inserire le rispettivi API key utilizzate per usufruire dei servizi di “captcha farm” trattati in precedenza, il ritardo che indica il tempo in secondi in cui il software ripete una richiesta dopo un tentativo fallito e infine il link webhook che rappresenta l'identificativo univoco di un server sulla piattaforma Discord in cui mandare le notifiche in caso di acquisto con successo.

Di seguito vengono riportate le principali funzionalità del bot sviluppato.

3.1 Sistema di autenticazione

Il sistema di autenticazione nasce dalla necessità di testare il sistema. Infatti, nel corso degli ultimi sei mesi il prodotto è stato rilasciato in forma beta ad un numero ristretto di utenti in modo tale che quest'ultimi potessero valutare il corretto funzionamento del software e in caso contrario segnalare eventuali bug. L'autenticazione è basata su una chiave univoca per ogni beta tester, sul nome dell'host con cui si accede al bot e infine sul proprio indirizzo IP. Tramite questi tre parametri viene effettuata una richiesta al mio server in cui è presente un demone in ascolto delle richieste in grado di restituire

uno status code alla richiesta effettuata a seconda se la chiave non esiste, è già utilizzata, oppure se attiva su un altro dispositivo, permettendo in questo modo l'utilizzo di una singola istanza per dispositivo per utente.

3.2 Sistema di log delle operazioni

Al fine di migliorare la praticità nel riscontrare errori o risposte non previste durante l'utilizzo del software, quest'ultimo è stato dotato di un sistema di log delle operazioni in grado di scrivere sul file "debug.log" tutte le richieste effettuate, le eventuali risposte e i singoli errori. L'intero sistema di log è stato facilmente sviluppato utilizzando la libreria di logging presente nativamente nel linguaggio adottato: Python [39].

3.3 Il sistema di notifiche

Una volta completato il processo di acquisto, quest'ultimo sarà notificato attraverso una notifica sulla nota piattaforma Discord.

“Discord integra la funzione dei webhooks come sistema intuitivo di automatizzazione di messaggi e aggiornamenti inviati ad un canale testuale del tuo server. Pensa ai webhooks come quei tubi pneumatici in cui inserisci i soldi in una banca e li guardi scomparire risucchiati dentro al tubo, solo che invece che non vedere mai più i tuoi soldi, stai inviando messaggi su Discord tramite un'altra piattaforma. [40]”

“Creando il punto di arrivo di un webhook nel tuo server, verrà generato un URL Webhook che potrai inserire in un buon numero di altri servizi per collegare i due [41]”

In questo modo, l'utente potrà visualizzare la taglia e il prodotto acquistato con un relativo link per effettuare manualmente il pagamento. Questo link si rende necessario dal momento in cui i siti supportano esclusivamente pagamento tramite PayPal oppure

nel caso in cui supportino anche pagamento tramite carta di credito o di debito sarà necessario confermare il codice di verifica tramite browser. La notifica, varia a seconda del sito, ma nel caso generale presenta le seguenti informazioni:

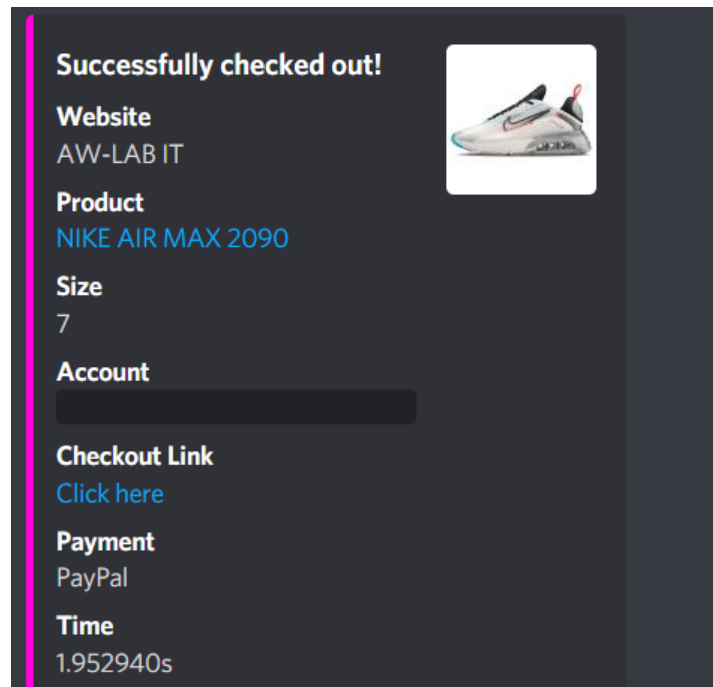


Fig. 3.6 Esempio di notifica generata in seguito all’acquisto sul sito AW-LAB

3.3 Modalità “shock drop”

Come suggerito dal nome, questa modalità è stata sviluppata per i negozi virtuali che spesso effettuano rilasci a sorpresa. In questo modo, i server riceveranno un traffico minore riservato alle persone a conoscenza del rilascio a sorpresa. L’algoritmo è sviluppato sulla base del riconoscimento dell’URL.

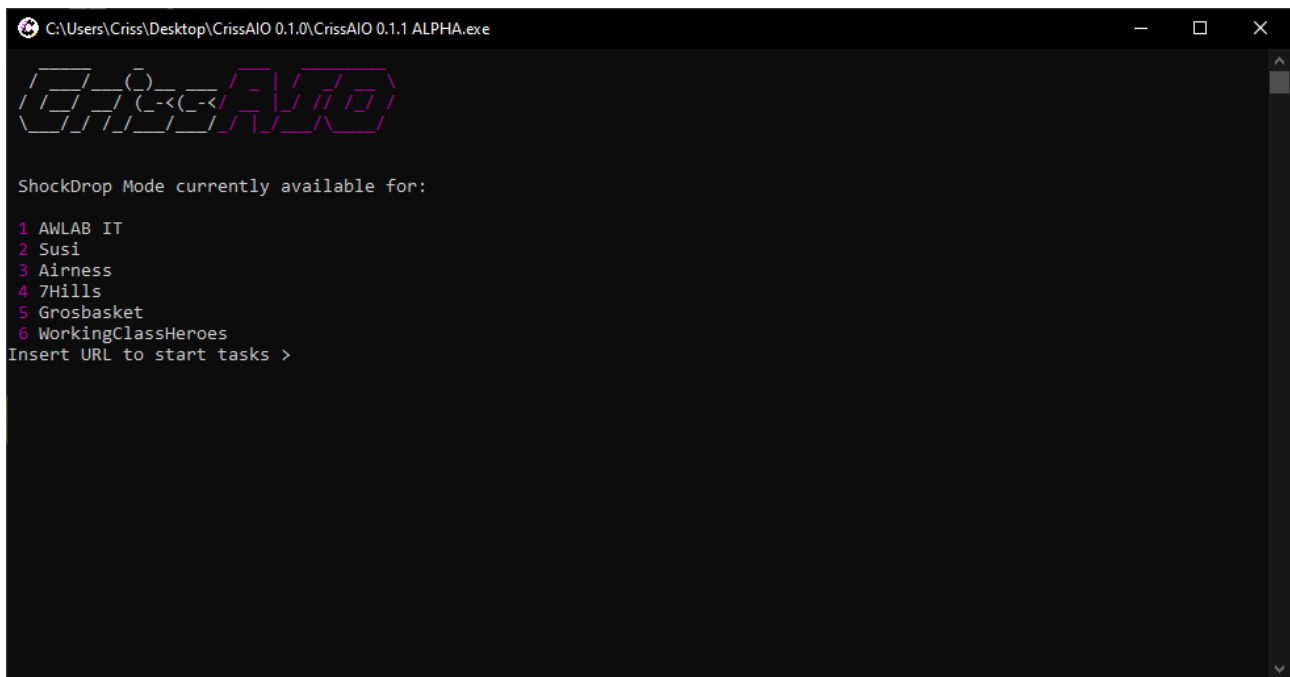


Fig. 3.7 Interfaccia sviluppata per la modalità “shock drop”

Una volta inserito l’URL del prodotto, l’algoritmo provvederà a riconoscere il negozio online e avvierà tante task quanto il numero specificato nel file config.json utilizzando il primo profilo disponibile nel file task.csv relativo al negozio inserito.

3.4 Gestione del pagamento

Una volta completato l’ordine il programma manderà una notifica che permetterà all’utente di pagare il prodotto selezionato. Inizialmente, la modalità di pagamento si basava su Selenium, un tool per l’automazione di browser. Attraverso il driver di Chrome, viene aperta una pagina relativa al pagamento tramite carta di credito o PayPal e verranno caricati i cookies relativi alla sessione di acquisto in modo tale che il pagamento rifletta realmente il prodotto aggiunto al carrello.

```

driver = webdriver.Chrome('data/chromedriver.exe')
driver.get('https://paypal.com')
driver.delete_all_cookies()
for c in s.cookies:
    z = {'name':c.name, 'value':c.value, 'path':c.path}
    driver.add_cookie(z)

driver.get(pay.url)

```

Fig. 3.8 Algoritmo per finalizzare il pagamento utilizzando Selenium

Al fine di rendere il pagamento più veloce e non sovraccaricare troppo le risorse di sistema aprendo numerose schermate di Chrome è stata sviluppata un'estensione Chrome in JavaScript grazie alla quale aprendo il link direttamente nella notifica sarà possibile caricare tutti i cookies direttamente nel proprio browser Chrome.

3.5 Le piattaforme supportate

Prima di trattare delle piattaforme e i conseguenti siti e-commerce supportati dal software, verranno introdotte le istruzioni ripetute per ogni richiesta POST o GET effettuata.

Infatti, tutte le richieste effettuate ai diversi siti sono accumulate da una serie di controlli effettuate sulle risposte. Di ogni response, infatti, viene controllato lo status code in modo tale che se quest'ultimo equivale a "503", valore che indica il ban temporaneo dell'indirizzo, verrà prelevato automaticamente un nuovo proxy dal file apposito.

```

if str(r.status_code) == '503':
    print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..')
    logging.info('[AIRNESS] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
    loadProxies()
    continue

```

Fig. 3.9 Istruzioni in merito allo status code della response in seguito ad una richiesta

Inoltre, per effettuare le richieste ai server viene utilizzata la libreria Requests disponibile per Python, la quale integra una serie di eccezioni in caso la richiesta non vada a buon fine.

```
▼ except requests.exceptions.ProxyError as errp:
    print(threadTime() + bcolors.FAIL + '[PROXY ERROR] Proxy banned, retrying with a new one..')
    logging.info('[AIRNESS] ' + threadTime() + '[' + str(errp) + '] [PROXY ERROR] Proxy banned, retrying with a new one..')
    loadProxies()
    time.sleep(delay)
    continue
▼ except requests.exceptions.ConnectionError as errh:
    print(threadTime() + bcolors.FAIL + '[CONNECTION ERROR] Proxy banned, retrying with a new one..')
    logging.info('[AIRNESS] ' + threadTime() + '[' + str(errh) + '] [CONNECTION ERROR] Proxy banned, retrying with a new one..')
    loadProxies()
    time.sleep(delay)
    continue
▼ except requests.exceptions.Timeout as errt:
    print(threadTime() + bcolors.FAIL + '[TIMEOUT] Failed checking out, retrying..' + bcolors.ENDC)
    logging.info('[AIRNESS] ' + threadTime() + '[' + str(errt) + '] [TIMEOUT] Failed checking out, retrying..')
    time.sleep(delay)
    continue
▼ except Exception as e:
    print(threadTime() + bcolors.FAIL + '[ERROR] Failed checking out, retrying..' + bcolors.ENDC)
    logging.info('[AIRNESS] ' + threadTime() + '[' + str(e) + '] [ERROR] Failed checking out, retrgying..')
    time.sleep(delay)
    continue
```

Fig. 3.10 Exception supportate dal modulo request in seguito ad una richiesta

Successivamente verranno mostrati alcuni dei siti supportati in base alle loro peculiarità.

3.5.1 Modulo Demandware

Attualmente il software supporta due siti basati sulla piattaforma Demandware. Data la somiglianza delle richieste, ne verrà riportato solo uno a titolo di esempio. L'e-commerce trattato è Susi, un negozio che ha sede a Latina e che spesso rilascia prodotti a tiratura limitata della casa adidas.

Il file relativo alla task prevede il link del prodotto di cui si desidera effettuare l'acquisto, la taglia che può essere una taglia specifica, una range di taglie desiderate o una taglia scelta casualmente, e le informazioni relative all'indirizzo di spedizione. Infine, il pagamento sarà effettuato esclusivamente tramite PayPal, unico sistema di pagamento supportato dal sito in questione.

Una volta avviato, l’algoritmo provvederà, per ogni riga formattata correttamente, ad eseguire una richiesta alla pagina del prodotto desiderato in modo tale da prelevare informazioni sul nome del prodotto, l’immagine e le taglie disponibili. In particolare, dalla sorgente HTML della pagina è possibile controllare le taglie disponibili cercando tutti gli elementi ancora definiti sotto la classe “swatchanchor”.

```
<li class="selectable">
  <a class="swatchanchor"
    href="https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/Product-Variation?pid=ADIDAS%2e-EF0347&dwvar_ADIDAS%2e-EF0347_size=292&dwvar_ADIDAS%2e-EF0347_color=86797"
    title="Seleziona Taglia: 7.5"
    data-igimg="{&quot;url&quot;:&quot;/on/demandware.static/-/Sites-susi-catalog-master/default/dwcd09fec2/images/large/2000215675770/200021567577001.jpg&quot;,&quot;title&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK&quot;,&quot;alt&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK, large&quot;,&quot;hires&quot;:&quot;}&quot;}">
    7.5
  </a>
</li>

<li class="selectable">
  <a class="swatchanchor"
    href="https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/Product-Variation?pid=ADIDAS%2e-EF0347&dwvar_ADIDAS%2e-EF0347_size=293&dwvar_ADIDAS%2e-EF0347_color=86797"
    title="Seleziona Taglia: 8"
    data-igimg="{&quot;url&quot;:&quot;/on/demandware.static/-/Sites-susi-catalog-master/default/dwcd09fec2/images/large/2000215675770/200021567577001.jpg&quot;,&quot;title&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK&quot;,&quot;alt&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK, large&quot;,&quot;hires&quot;:&quot;}&quot;}">
    8
  </a>
</li>
```

Fig. 3.11 HTML relativo ad un prodotto caricato su Susi

A questo punto gli identificativi per aggiungere il prodotto al carrello sono ottenibili eseguendo una richiesta all’indirizzo presente nel tag “href” e inserendo i parametri “&Quantity=1&format=ajax&productlistid=undefined”. Una volta ottenute tutte le informazioni riguardanti il prodotto sarà possibile aggiungerlo al carrello. Tale operazione viene effettuata tramite una richiesta POST all’indirizzo: https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/Cart-AddProduct?format=ajax

E inserendo nel body della richiesta:

```
Quantity: 1
cartAction: add
pid: 2235776
```

Fig. 2.11 Body richiesta POST all’indirizzo https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/Cart-AddProduct?format=ajax

Dove pid è l’identificatore della taglia selezionata in precedenza. Poiché il sito prevede che la richiesta sia XMLHTTP, ovvero un set di api che permettono di trasferire XML da e vero server web tramite http, l’header della richiesta conterrà tale opzione.

```
headersXML = {
  'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.162 Safari/537.26',
  'x-requested-with': 'XMLHttpRequest'
}
```

Fig. 2.12 Esempio di header per effettuare una richiesta XML

Inoltre, il sito utilizza un CSRF token. Un CSFR token è un valore unico, segreto e non predicibile che viene generato da una applicazione lato server e trasmesso al cliente in modo tale da essere incluso in una successiva richiesta HTTP effettuata dal cliente. Quando successivamente viene effettuata una richiesta dal cliente, l'applicazione lato server convalida che tale richiesta includa il token aspettato o rifiuti le richieste se il token manca o non è valido [42].

Nel caso di Susi, è possibile ottenere il token in questione semplicemente effettuando una richiesta all'indirizzo del carrello e ricercando il token all'interno della pagina HTML restituita.

```
</div>
</div>
<input type="hidden" name="csrf_token" value="i4E6SgrHtpT3NDfXDFEOW_BBE355Khly83bhAbnpsc4muRNATHZLRKLAfNfCMd1r32_bVK406hv33FI7he99B65h8xp0nMBUZH5r_XGKrZbvmUPGTcvqf6s8c0j78HLa91vr9-DK9TEIIPm-iHM2IL9Fv-
kpR1vPbrmVrBe-BP78e5Dpk="/>
</fieldset>
</form>
<div class="cart-actions">
```

Fig. 2.13 Pagina HTML relativa al token CSFR

Successivamente viene selezionato il checkout come ospite e inviati i dati relativi all'indirizzo di spedizione e di fatturazione tramite le seguenti richieste:

```
Request URL: https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/COCustomer-LoginForm?scope=checkout
Request Method: POST
Status Code: 302
```

Fig. 2.14 Richiesta POST sul sito Susi per effettuare il checkout come ospite

```
dwfrm_login_unregistered: Checkout come ospite
csrf_token: FFayM3kxy9xQi0ahZCHf00dqmnl1s-9gY1reBZhGjnJb3bN9dyvJ8uYSN7Le8b0Pk991HVuGF14Dz0tX7hxj-dV9JGIU3iMikEG7FA4U
uho7k5U7xtYy_60_QQApdx-9d1ovb6bLHf1WB5ZJDSzogsBmPtgPLngBE24mnMhfcNOBtZDs=
```

Fig. 2.15 Body richiesta POST sul sito Susi per effettuare il checkout come ospite

```
Request URL: https://www.susi.it/it-IT/cliente/fatturazione/
Request Method: POST
Status Code: 200
```

Fig. 2.16 Richiesta POST sul sito Susi per caricare i dati relativi alla spedizione

```
dwfrm_singleshipping_shippingAddress_addressFields_firstName:
dwfrm_singleshipping_shippingAddress_addressFields_lastName:
dwfrm_singleshipping_shippingAddress_addressFields_address1:
dwfrm_singleshipping_shippingAddress_addressFields_address2:
dwfrm_singleshipping_shippingAddress_addressFields_postal:
dwfrm_singleshipping_shippingAddress_addressFields_city:
dwfrm_singleshipping_shippingAddress_addressFields_phone:
dwfrm_singleshipping_shippingAddress_addressFields_country:
dwfrm_singleshipping_shippingAddress_addressFields_states_state:
dwfrm_singleshipping_shippingAddress_shippingMethodID: DHL-Express
dwfrm_singleshipping_shippingAddress_save: Passa a Fatturazione >
csrf_token: xIxGEjgvbUVxSwA2j_BIFcdR-_xAZr7ZSFIjILQrrpHP51y3mOGmf27CvA11DHSRL18GD418VYzg5fvHvoBGOCcM1PrSeOD9dcnPdgpN
c80b_1Ve4vFS8goWoKhcRC3oItYjdUXivE0cdfemX5_cwFp2yieITm-etmJDGHqBj8pS5RuI_0=
```

Fig. 2.17 Body richiesta POST sul sito Susi per caricare i dati relativi alla spedizione

Una volta inserite correttamente le informazioni di spedizione e fatturazione, sarà possibile visualizzare dalla pagina di riepilogo d'ordine, l'elemento utilizzato per il pagamento e relativo completamento d'ordine.

Fig. 2.19 Frame per il pagamento disponibile all'indirizzo <https://www.susi.it/it-IT/cliente/riepilogo-ordine/>

Ricaricando il frame relativo al pagamento è possibile catturare la richiesta relativa al pagamento PayPal.

```
Request URL: https://securepayments.paypal.com/webapps/HostedSoleSolutionApp/webflow/sparta/hostedSoleSolutionProcess?test_ipn=1&
Request Method: POST
Status Code: 200 OK
```

Fig. 2.20 Richiesta POST sul sito Susi per caricare il frame relativo al pagamento

Infine, una volta selezionato il pagamento si potrà ottenere l'indirizzo univoco PayPal legato all'operazione effettuando una richiesta POST all'indirizzo, il quale, come si intuisce dal codice di ritorno 302, ritorna il link desiderato.

```
Request URL: https://www.paypal.com/cgi-bin/webscr?cmd=_hssnode-merchantpaymentweb
Request Method: POST
Status Code: 302
```

Fig. 2.21 Richiesta POST sul sito Susi per ottenere il link PayPal per il pagamento

Tale link se aperto non mostrerà il prezzo relativo al prodotto in quanto non sono presenti i cookies utilizzati lato server per mantenere le informazioni relative all'ordine. Attraverso la seguente funzione, verranno prelevati tutti i cookies relativi alla sessione corrente, codificati in base 64 e infine inseriti nell'URL disponibile direttamente dalla notifica. Tale procedura viene applicata per tutti i siti e tutte le piattaforme in cui non è previsto il completamento dell'ordine in maniera completamente automatizzata.

```

def getCheckoutLink(s, url, checkout):
    elem = '{ "url":"' + url + '", "checkout":"' + checkout + '" }'
    newJSON = json.loads(elem)

    cookieJSON = []
    for c in s.cookies:
        #print(c)
        elem = '{ "name":"' + c.name + '", "domain":"' + c.domain + '", "path":"' + c.path + '", "value":"' + c.value + '" }'
        elemJSON = json.loads(elem)
        cookieJSON.append(elemJSON)

    newJSON.update({'cookies': cookieJSON})

    finalCheckout = str(newJSON)
    message_bytes = finalCheckout.encode('ascii')
    base64_bytes = base64.b64encode(message_bytes)
    base64_message = base64_bytes.decode('ascii')

    all_url = 'https://crissaio.com/checkout?data='+base64_message

    try:
        obj = UrlShortenTinyurl()
        shortURL = obj.shorten(all_url)
    except Exception as e:
        traceback.print_exc()

    return shortURL

```

Fig. 2.22 Algoritmo per la generazione dell'URL contenente i cookie necessari a finalizzare l'acquisto

Di seguito viene riportato a titolo d'esempio un acquisto di due prodotti diversi:

- https://www.susi.it/it-IT/uomo/scarpe/sneaker/scarpa-free-hiker-parley/ADIDAS.-EF0347.html?dwvar_ADIDAS.-EF0347_color=86797&cgid=803
- https://www.susi.it/it-IT/uomo/scarpe/sneaker/scarpa-supercourt/ADIDAS.-EE6038.html?dwvar_ADIDAS.-EE6038_color=86797&cgid=803

```

C:\Users\Criss\Desktop\CrissAIO 0.1.0\CrissAIO 0.1.0.exe
[17:08:03.620123] [Task-1] No proxies found, running localhost..
[17:08:03.621123] [Task-1] Getting product info..
[17:08:03.621123] [Task-2] No proxies found, running localhost..
[17:08:03.622131] [Task-2] Getting product info..
[17:08:04.993469] [Task-2] Size available: [6.5] [7] [7.5] [8] [8.5] [9] [9.5] [10] [10.5]
[17:08:04.993469] [Task-2] Adding to cart size: [9.5]
[17:08:05.136147] [Task-2] [200] Successfully added to cart!
[17:08:05.136147] [Task-2] Getting checkout token..
[17:08:05.182015] [Task-1] Size available: [6.5] [7] [7.5] [8] [8.5] [9] [9.5] [10] [10.5] [11]
[17:08:05.183168] [Task-1] Adding to cart size: [9.5]
[17:08:05.351737] [Task-1] [200] Successfully added to cart!
[17:08:05.351737] [Task-1] Getting checkout token..
[17:08:05.382952] [Task-2] [200] Successfully got checkout token!
[17:08:05.434132] [Task-2] Selecting guest checkout..
[17:08:05.605462] [Task-1] [200] Successfully got checkout token!
[17:08:05.678694] [Task-1] Selecting guest checkout..
[17:08:06.031980] [Task-2] [200] Successfully selected guest checkout!
[17:08:06.032994] [Task-2] Checking out..
[17:08:06.202705] [Task-1] [200] Successfully selected guest checkout!
[17:08:06.202705] [Task-1] Checking out..
[17:08:06.679453] [Task-2] Getting paypal link..
[17:08:07.000344] [Task-1] Getting paypal link..
[17:08:13.833437] [Task-2] [200] Successfully checked out!
[17:08:13.946879] [Task-1] [200] Successfully checked out!

```

Fig. 2.23 Acquisto di due prodotti utilizzando il bot sviluppato su Susi

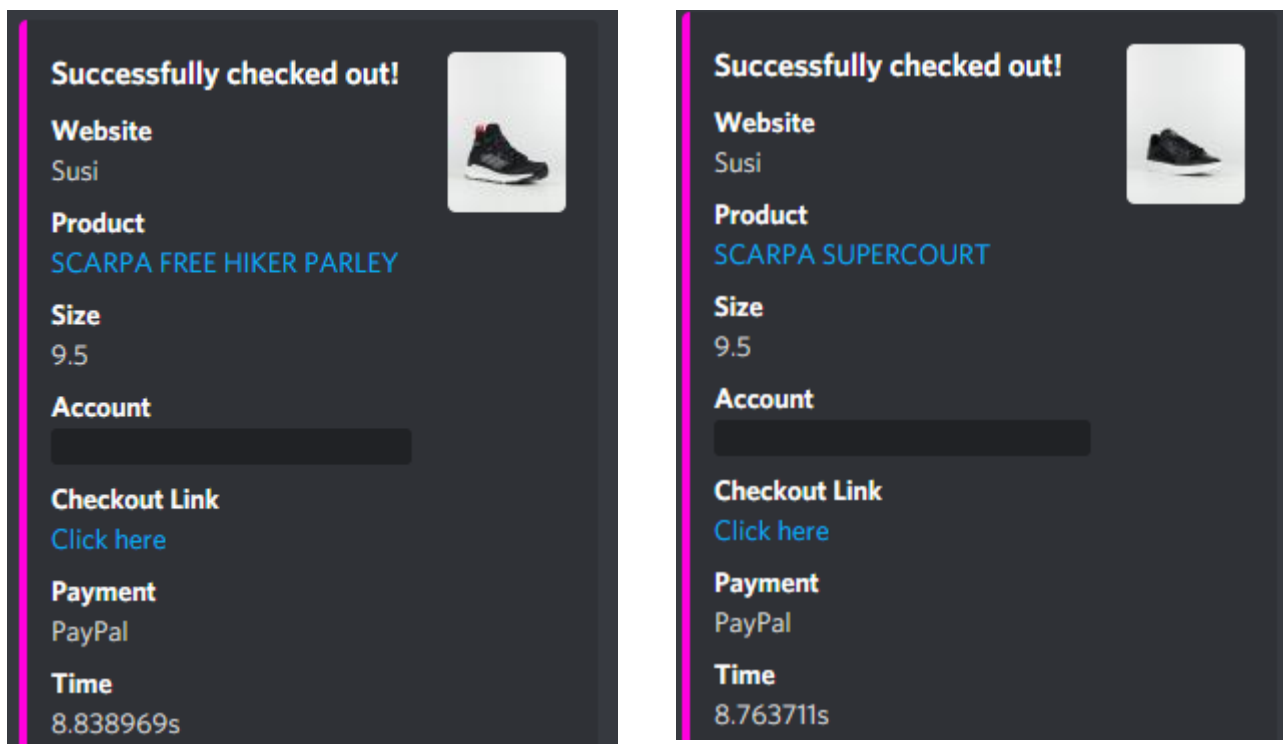


Fig. 2.24 / 2.25 Notifica webhook in seguito all'acquisto del prodotto desiderato su Susi

3.5.2 Modulo Magento

Data la popolarità della piattaforma Magento, la maggior parte degli e-commerce supportati dal software sono basati su questa e-platform. Di seguito verranno discussi due siti, entrambi protetti dalla tecnologia CloudFlare precedentemente trattata, i quali si distinguono in base al livello di sicurezza applicata. La scelta ricade su Holypopstore disponibile all'indirizzo <https://www.holypopstore.com/it>, situato su un server CloudFlare. Infatti, effettuando qualsiasi richiesta al sito, si potrà estrapolare dall'header della risposta come quest'ultima provenga da un server CloudFlare.

```
Request URL: https://www.holypopstore.com/it/footwear-uomo/adidas/adidas-gazelle-vintage/10889
Request Method: GET
Status Code: 200
Remote Address: 104.26.6.54:443
Referrer Policy: strict-origin-when-cross-origin

Response Headers
cache-control: no-cache, max-age=0, must-revalidate, no-store
cf-cache-status: DYNAMIC
cf-ray: 5d0a231d39d7e8fb-MXP
cf-request-id: 051a34463e0000e8fbcd398200000001
content-encoding: br
content-type: text/html; charset=utf-8
date: Thu, 10 Sep 2020 15:21:10 GMT
expect-ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
server: cloudflare
```

Fig. 2.26 Header della response in seguito ad una richiesta effettuata al dominio <https://www.holypopstore.com/it>

Nonostante CloudFlare sia utilizzato come servizio di host per l'e-commerce, il livello di protezione adottato è relativamente basso. Infatti, per non essere rintracciati e di conseguenza basterà utilizzare per ogni richiesta un header correttamente formattato e un proxy unico per ogni sessione di acquisto, in modo tale che le operazioni eseguite appaiano al sito come se diversi utenti reali facessero richieste da diversi indirizzi IP.

Una peculiarità del sito è che una volta eseguito il login e aggiunto al carrello il prodotto desiderato, quest'ultimo sarà riservato per un acquisto successivo. Di conseguenza la prima operazione da effettuare sarà quella di accesso con il proprio account. La richiesta viene effettuata tramite una POST al seguente indirizzo

```
Request URL: https://www.holypopstore.com/index.php
Request Method: POST
Status Code: 200
```

Fig. 2.27 Richiesta POST sul sito Holypopstore per effettuare l'accesso


```
controller: auth
action: authenticate
type: standard
extension: holypop
credential:
password:
version: 346
cookieVersion: 310
```

Fig. 2.27 Body richiesta POST sul sito Holypopstore per effettuare l'accesso

La risposta dal parte del server conterrà le seguenti informazioni in formato Json.

Headers	Preview	Response	Initiator	Timing	Cookies
1	{ "success": true, "payload": "94adefc2168300a50701077191d345a1" }				

Fig. 2.27 Body response sul sito Holypopstore in seguito alla richiesta di accesso

Come si può vedere, analizzando la risposta e caricandola in un oggetto Json è possibile verificare il successo dell'operazione semplicemente estrapolando l'elemento "success" dalla risposta. Una volta eseguito il login sarà possibile ricercare il relativo "AddressID" utilizzato come informazione da inviare al server nel momento del caricamento delle informazioni di spedizione. Per farlo basterà eseguire una richiesta GET all'indirizzo "https://www.holypopstore.com/en/account/address-book" e cercando all'interno del codice HTML è possibile ritrovare il valore cercato sotto la variabile Javascript "preloadedAddressBook". A questo punto verrà effettuata una richiesta alla pagina del prodotto desiderato al fine di prelevare direttamente dal codice HTML le informazioni riguardanti il titolo, l'immagine e le taglie utilizzate successivamente per aggiungere l'articolo al carrello e per formattare la notifica da inviare in caso di acquisto effettuato con successo. In particolare la ricerca delle taglie è un processo che richiede tempo, di conseguenza sono state sviluppate due modalità selezionabili direttamente dal file csv delle task:

- “SAFE”: la modalità SAFE permette di prelevare tutte le taglie disponibili direttamente dalla pagina HTML. Questa procedura rende l’algoritmo più lento in quanto si andrà a prelevare la variabile in cui sono salvate tutte le informazioni e a elaborarla in modo da estrapolare tutti i variant necessari per aggiungere il prodotto al carrello.

```

if mode == 'FAST':
    break
else:
    ListOfRawSizes = list(pPage.find_all('script'))

    for line in ListOfRawSizes:
        if 'preloadedStock' in str(line):

            ListOfRawSizes = str(ListOfRawSizes).split('var preloadedStock = ')[1].split(';')[0] + ']'
            #print(ListOfRawSizes)
            stockJson = json.loads(ListOfRawSizes)
            #print(stockJson)

            for id in stockJson:
                #print(id)
                pid_size = str(id).split('stockItemId')[1].split(',')[0].replace("'", '').strip().replace(':', '').strip().replace('u', '').strip()
                #print(pid_size)
                size = str(id).split('variant')[1].split(',')[0].split(':')[1].replace("'", '').strip().replace('u', '').strip()
                #print(size)
                in_stock = str(id).split('isInStock')[1].split(' ')[1].split('')[0].replace("'", '').strip()
                #print(in_stock)
                if in_stock:
                    sizeDict.update({pid_size:size})

```

Fig. 2.28 Algoritmo sviluppato per estrapolare le informazioni relative al prodotto su Holypopstore

- “FAST”: la modalità FAST è utilizzata quando un prodotto è stato precedentemente caricato sul sito. Infatti, essa è basata sull’utilizzo diretto degli identificativi delle taglie, che permettono di aggiungere al carrello il prodotto direttamente senza estrapolare informazioni dalla pagina HTML. Tale modalità da una parte è più veloce ma dall’altra sacrifica l’eventualità che la taglia non sia disponibile. Infatti, se in modalità SAFE vengono controllate le taglie disponibile direttamente dalla pagina del prodotto, la modalità FAST sacrifica questa possibilità al fine di guadagnare tempo fondamentale in presenza di un’alta concorrenza. Al fine di garantire una maggiore facilità di utilizzo agli utenti, nel caso si utilizzi questa modalità, i variant delle taglie vengono messi a disposizione in un apposito canale.

In seguito viene eseguita l’operazione di aggiunta al carrello tramite una richiesta di POST con uno specifico corpo.

Request URL: https://www.holypopstore.com/index.php
Request Method: POST
Status Code: 200

Fig. 2.29 Richiesta POST sul sito Holypopstore per aggiungere al carrello un prodotto

▼ **Form Data** view source view URL encoded

controller: orders
action: addStockItemToBasket
stockItemId: 55739
quantity: 1
extension: holypop
version: 346
cookieVersion: 310

Fig. 2.29 Body richiesta POST sul sito Holypopstore per aggiungere al carrello un prodotto

Dove stockItemId è il variant della taglia desiderata. Anche in questo caso, analizzando la risposta alla richiesta è possibile verificare l'esito dell'operazione.

```
{
  "success": true,
  "payload": [
    {
      "id": 2591447, "itemId": "10709", "modelCode": "CJ6779-100", "stockId": "55739", "code": "194276554838",
      "alias": "nike-air-max-iii", "title": "Air Max III", "subtitle": "white \\/ black-grey fog-laser blue",
      "permalink": "https:\\\\www.holypopstore.com\\/it\\/footwear-uomo\\/nike\\/nike-air-max-iii\\/10709",
      "category": {
        "id": "4", "title": "Uomo", "permalink": "https:\\\\www.holypopstore.com\\/it\\/footwear-uomo"
      },
      "manufacturer": {
        "id": "69", "title": "Nike", "permalink": "https:\\\\www.holypopstore.com\\/it\\/nike"
      },
      "attributes": [
        {
          "attribute": {
            "id": "3", "title": "Taglia", "type": "DEFAULT", "group": "SIZE"
          },
          "value": {
            "id": "13", "title": "9 US", "alias": "9-us", "displayName": " 9 US", "attributeId": "3",
            "value": "9 US", "altvalue": null, "ordering": "25"
          }
        }
      ],
      "displayAttributeValues": ["9 US"],
      "imageObject": {
        "id": "21917", "galleryId": "10118", "name": "airmax3ok222.png", "extension": "png",
        "path": "\\images\\/galleries\\/10118", "url": "https:\\\\data.holypopstore.com\\/images\\/galleries\\/10118\\/airmax3ok222.png",
        "imageUrl": "https:\\\\data.holypopstore.com\\/images\\/galleries\\/10118\\/airmax3ok222.png",
        "thumbUrl": "https:\\\\data.holypopstore.com\\/images\\/galleries\\/10118\\/thumbs\\/airmax3ok222.png",
        "width": "2000", "height": "2000", "aspectRatio": "1.00", "title": "", "caption": "",
        "altTitle": "Air Max III white \\/ black-grey fog-laser blue", "link": "", "ordering": "0",
        "contentType": "IMAGE", "type": "IMAGE", "versions": {
          "thumbnails": "https:\\\\data.holypopstore.com\\/images\\/galleries\\/10118\\/thumb",
          "full": "https:\\\\data.holypopstore.com\\/images\\/galleries\\/10118\\/airmax3ok222.png",
          "0": "https:\\\\data.holypopstore.com\\/ima",
          "1": "https:\\\\data.holypopstore.com\\/images\\/galleries\\/10118\\/cover\\/airmax3ok222_v1.png",
          "set": "https:\\\\data.holypopstore.com\\/images\\/galleries\\/10118\\/cover\\/airmax3ok222_v1.png"
        },
        "availableFrom": "2020-08-01 13:00:00", "isInPreOrder": 0, "pricelistId": null, "weight": "0.000", "price": "151.00", "taxedPrice": 151, "currency": {
          "id": "33", "code": "EUR", "iso": "978", "symbol": "&euro;", "status": "ACTIVE", "rate": "1.0000"
        },
        "storesList": [
          {
            "id": "1", "isCustom": false, "isGift": false, "mode": "SELL", "quantity": "1", "oldQuantity": 0, "addedQuantity": "1", "times": 1, "thereWereChangesToItem": false, "discountType": null, "discount": null
          }
        ]
      }
    ]
  ]
}
```

Fig. 2.30 Json in risposta alla richiesta di aggiungere al carrello di un prodotto su Holypopstore

Come detto in precedenza, una volta aggiunto al carrello, il prodotto sarà riservato e potrà essere acquistato direttamente accedendo con l'account utilizzato. Tuttavia, l'ordine viene completato eseguendo le seguenti operazioni:

```

checkout_payload = {
    'secretly': 'false',
    'hardErrorize': 'false',
    'billingAddressId': shippingAddressId,
    'shippingAddressId': shippingAddressId,
    'shipmentFlow': 'DELIVERY',
    'newAddresses': 0,
    'requestInvoice': 0,
    'notes': '',
    'paymentMethodId': 1,
    'paymentMethodAccountId': 1,
    'shipments[0][shipmentFlow]': 'DELIVERY',
    'shipments[0][addressId]': shippingAddressId,
    'shipments[0][shipperId]': 10,
    'shipments[0][shipperAccountId]': 4,
    'toDisplay': 0,
    'extension': 'holypop',
    'controller': 'orders',
    'action': 'save',
    'language': 'EN',
    'version': 294,
    'cookieVersion': 294
}

```

Fig. 2.31 Dizionario che rappresenta il payload della richiesta POST di completare l'ordine su Holypopstore

```

r = session.post('https://www.holypopstore.com/index.php', headers=headersXML, data=checkout_payload, proxies=proxy)

if str(r.status_code) == '503':
    print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..' + bcolors.ENDC)
    logging.info('[HOLYPOP] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
    loadProxies()
    continue

```

Fig. 2.31 Istruzioni relative alla richiesta POST per completare l'ordine su Holypopstore

Dalla risposta sarà possibile estrapolare l'orderId utilizzato nell'ultima POST da cui si ottiene, come indirizzo di ritorno, il link per il pagamento tramite PayPal.

```

link = 'https://www.holypopstore.com/it/orders/checkout/' + str(orderId) + '?paymentMethodId=1&paymentMethodAccountId=1'

while True:
    try:
        r = session.get(link, headers=headers, proxies=proxy)

        if str(r.status_code) == '503':
            print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..' + bcolors.ENDC)
            logging.info('[HOLYPOP] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
            loadProxies()
            continue
    
```

Fig. 2.31 Istruzioni per ottenere l'indirizzo di pagamento PayPal su Holypopstore

Il secondo sito analizzato per la categoria Magento è Grosbasket. La scelta ricade non solo per la miriade di rilasci di calzature Jordan ma anche per la forte protezione CloudFlare scelta dai proprietari. Infatti, accedendo al sito, sarà possibile incontrare la tipica schermata di attesa CloudFlare. Questo comporta la risoluzione di una sfida all'interno della pagina. Tale operazione è lasciata alla libreria CloudFlare disponibile su GitHub e sviluppata da VeNoMouS [43]. Essa verrà analizzata nel capitolo successivo

e integra la normale libreria Request al fine di mantenere le stesse funzioni nonostante la presenza del sistema di protezione.

Di conseguenza la prima operazione da effettuare sarà la risoluzione della sfida CloudFlare tramite semplicemente una richiesta GET all'indirizzo del prodotto utilizzando la specifica libreria. Successivamente verranno prelevate tutte le informazioni riguardanti il prodotto.

```
soup = BeautifulSoup(r.text.encode('utf-8'), 'html.parser')

productTitle = str(soup.find('li', {'class': 'product'}).text).replace('\n', '')
productIMG = soup.find('img', {'alt': productTitle})['src']
form_key = soup.find('input', {'name': 'form_key'})['value']
product_id = soup.find('input', {'name': 'product'})['value']
category = str(soup.find('li', {'class': 'nav-item level2 nav-1-3-10 classic'})).split('<a href=""')[1].split('>')[0]
super_attr = soup.find('select', {'class': 'required-entry super-attribute-select'})['name']
```

Fig. 2.32 Istruzioni per ottenere le informazioni riguardanti il prodotto su Grosbasket

```
soup = BeautifulSoup(r.text.encode('utf-8'), 'html.parser')

pidInStock = str(soup).split('var switcherConfig=')[1].split(', "images:"')[0]+'}'
pidInStockJson = json.loads(pidInStock)['stock']

size = str(soup).split('Product.Config(')[1].split(');</script>')[0]
temp = '['+size.split('[')[1].split(']')[0]+'}'
temp = list(temp.split(","))

sizeDict = {}

for line in temp:
    #print(line)
    pid = line.split('"id:"')[1].split(",")[0]
    #print(pid)
    size = line.split('"label:"')[1].split(",")[0].replace('\u2154', ' 2/3').replace('\u2153', ' 1/3')
    #print(size)
    productID = line.split('"products:"')[1].split(",")[0]
    #print(productID)
    if(pidInStockJson[productID]):
        sizeDict.update({pid:size})
```

Fig. 2.33 Istruzioni per ottenere le informazioni riguardanti le taglie di un prodotto su Grosbasket

Una volta ottenute le informazioni relative alle taglie, sarà possibile aggiungere il prodotto al carrello effettuando la seguente richiesta:

```
Request URL: https://www.grosbasket.com/it/amcart/ajax/index/
Request Method: POST
Status Code: 200
```

Fig. 2.34 Richiesta POST per aggiungere un prodotto al carrello su Grosbasket

```
product_id: 6244
form_key: rUgToxN1S4Bgt1su
product: 6244
related_product:
super_attribute[140]: 23
qty: 1
IsProductView: 1
current_category: https://www.grosbasket.com/it/scarpe/kolekcija-zvezdnikov/mic
hael-jordan
```

Fig. 2.35 Body richiesta POST per aggiungere un prodotto al carrello su Grosbasket

Analizzando la risposta, sarà possibile valutare se il prodotto è stato correttamente aggiunto al carrello.

```
{
  "title": "Informazione",
  "message": "<p>Hai aggiunto <a\nhref=\"https://www.grosbasket.com/it/air-jordan-super-fly-4\" title=\"Air Jordan 5
```

Fig. 2.36 Body response in seguito alla richiesta POST per aggiungere un prodotto al carrello su Grosbasket

A questo punto, per poter proseguire con l'ordine sarà necessario utilizzare un token necessario lato server al fine di riconoscere la sessione di acquisto per le operazioni future. Il token è facilmente prelevabile analizzando la sorgente HTML all'indirizzo <https://www.grosbasket.com/it/firecheckout/>. Successivamente saranno inviate le informazioni relative alla spedizione, prelevate dal file task.csv.

```

checkout_payload = {
    'billing[address_id]': address_id,
    'billing[firstname]': accountDict['first_name'],
    'billing[lastname]': accountDict['last_name'],
    'billing[email]': accountDict['email'],
    'billing[telephone]': accountDict['phone_number'],
    'billing[street][]': accountDict['address'],
    'billing[postcode]': accountDict['zipcode'],
    'billing[city]': accountDict['city'],
    'billing[country_id]': accountDict['country'],
    'billing[region_id]': '',
    'billing[region]': '',
    'shipping[same_as_billing]': '1',
    'billing[company]': '',
    'billing[vat_id]': '',
    'billing[customer_password]': '',
    'billing[confirm_password]': '',
    'billing[save_in_address_book]': '1',
    'billing[use_for_shipping]': '1',
    'shipping[address_id]': address_id,
    'shipping[firstname]': accountDict['first_name'],
    'shipping[lastname]': accountDict['last_name'],
    'shipping[telephone]': accountDict['phone_number'],
    'shipping[street][]': accountDict['address'],
    'shipping[postcode]': accountDict['zipcode'],
    'shipping[city]': '',
    'shipping[country_id]': accountDict['country'],
    'shipping[region_id]': '',
    'shipping[region]': '',
    'shipping[company]': '',
    'shipping[vat_id]': '',
    'shipping[save_in_address_book]': '1',
    'shipping_method': 'tablemate_bestway',
    'payment[method]': 'paypal_express',
    'coupon[remove]': '0',
    'coupon[code]': '',
    'order-comment': '',
    'cart_safe': cartAmount,
    'cartid': cartAmount,
    'payment[device_data]': deviceId,
    'review': '1',
}

```

Fig. 2.37 Dizionario che rappresenta il payload della richiesta POST per completare l'ordine su Grosbasket

```

r = s.post('https://www.grosbasket.com/it/firecheckout/index/saveOrder/form_key/'+form_key+'/', headers=headers, data=checkout_payload, proxies=proxy)

if str(r.status_code) == '503':
    print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..' + bcolors.ENDC)
    logging.info('[GROSBASKET] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
    loadProxies()
    continue

```

Fig. 2.38 Istruzione relative alla richiesta POST per completare l'ordine su Grosbasket

Infine, l'indirizzo per il pagamento sarà disponibile come URL di ritorno alla richiesta GET all'indirizzo <https://www.grosbasket.com/it/paypal/express/start/>. Di seguito si riporta il software che procede all'acquisto del prodotto <https://www.grosbasket.com/it/air-jordan-1-jester-xx-low-laced-wmns-black-white> taglia casuale con tre task.

```
C:\Users\Criss\Desktop\CrissAIO 0.1.0\CrissAIO 0.1.0.exe
[11:09:35.384026] [Task-1] Checking CloudFlare..
[11:09:35.403030] [Task-2] Checking CloudFlare..
[11:09:35.433037] [Task-3] Checking CloudFlare..
[11:09:35.726103] [Task-2] Getting product info..
[11:09:35.732104] [Task-3] Getting product info..
[11:09:35.734105] [Task-1] Getting product info..
[11:09:36.867795] [Task-3] Getting size info..
[11:09:36.946813] [Task-2] Getting size info..
[11:09:37.024830] [Task-1] Getting size info..
[11:09:37.137873] [Task-3] [200] Successfull got size info!
[11:09:37.137873] [Task-3] Size available: [36] [36.5] [37.5] [38] [38.5] [39] [40] [41] [42] [42.5]
[11:09:37.137873] [Task-3] Adding to cart size: [42]
[11:09:37.229893] [Task-2] [200] Successfull got size info!
[11:09:37.229893] [Task-2] Size available: [36] [36.5] [37.5] [38] [38.5] [39] [40] [41] [42] [42.5]
[11:09:37.229893] [Task-2] Adding to cart size: [36]
[11:09:37.320913] [Task-1] [200] Successfull got size info!
[11:09:37.321913] [Task-1] Size available: [36] [36.5] [37.5] [38] [38.5] [39] [40] [41] [42] [42.5]
[11:09:37.321913] [Task-1] Adding to cart size: [42]
[11:09:37.516027] [Task-3] [200] Successfully added to cart!
[11:09:37.517028] [Task-3] Getting checkout token..
[11:09:37.595045] [Task-2] [200] Successfully added to cart!
[11:09:37.595045] [Task-2] Getting checkout token..
[11:09:37.697006] [Task-1] [200] Successfully added to cart!
[11:09:37.698017] [Task-1] Getting checkout token..
[11:09:37.78527] [Task-3] [200] Successfully got checkout token!
[11:09:38.178527] [Task-3] Checking out..
[11:09:38.248544] [Task-2] [200] Successfully got checkout token!
[11:09:38.248544] [Task-2] Checking out..
[11:09:38.349565] [Task-1] [200] Successfully got checkout token!
[11:09:38.349565] [Task-1] Checking out..
[11:09:40.886757] [Task-1] [200] Successfully checked out!
[11:09:41.042227] [Task-2] [200] Successfully checked out!
[11:09:41.121999] [Task-3] [200] Successfully checked out!
```

Fig. 2.39 Acquisto di tre prodotti utilizzando il bot sviluppato su Grosbasket

3.5.3 Modulo WooCommerce

Per il modulo WooCommerce la scelta ricade su Airness, un e-commerce incentrato per lo più sulla vendita di prodotti inerenti al mondo del Basket. Tra questi, le ambite edizione limitate casa Jordan.

Durante il primo accesso viene generato un token, utilizzato lato server per individuare ogni accesso al sito e convalidare le operazioni successive. Infatti, una volta effettuata la seguente richiesta POST:

```
payload_token = {
    'scope': 'market:2392',
    'grant_type': 'client_credentials'
}
```

Fig. 2.40 Dizionario che rappresenta il payload della richiesta POST per ottenere il token su Airness


```

r = s.post('https://airness-2.commercelayer.io/oauth/token', headers=headers_token, data=payload_token, proxies=proxy)

if str(r.status_code) == '503':
    print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..')
    logging.info('[AIRNESS] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
    loadProxies()
    continue

tokenJson = json.loads(r.text)

```

Fig. 2.41 Istruzione relative alla richiesta POST per ottenere il token su Airness

Sarà possibile estrapolare dalla risposta in formato Json del server sia il token univoco che la durata in secondi di quest'ultimo. Successivamente ogni richiesta verso il server dovrà essere convalidata con un token valido e non scaduto. Per questo motivo ad ogni richiesta verrà valutata la risposta. Nel caso in cui quest'ultima contenga il messaggio di errore “The access token you provided is invalid”, sarà effettuata nuovamente la richiesta precedente al fine di procedere all'acquisto utilizzando un nuovo token corretto.

Per prelevare informazioni sui prodotti rilasciati, in questo caso non è sufficiente analizzare la pagina HTML del prodotto ma sarà necessario effettuare una richiesta POST all'indirizzo:

“https://cdn.contentful.com/spaces/40i19ww9637w/environments/master/entries”

Con le seguenti informazioni contenute nel body della richiesta:

```

params = {
    'content_type': 'node',
    'fields.slug[in]': productID,
    'fields.toRoot': 'true',
    'include': 10,
    'limit': 1,
    'locale': 'it-IT'
}

```

Fig. 2.42 Dizionario che rappresenta il payload della richiesta POST per ottenere informazioni sui prodotti su Airness

Analizzando la risposta a tale richiesta si potranno prelevare le informazioni riguardanti nome del prodotto, immagine e SKU (Stock Keeping Unit). Quest'ultimo utilizzato per ottenere le informazioni riguardanti le taglie attualmente disponibili tramite una richiesta di POST all'indirizzo

```
'Link', 'linkType': 'ContentType', 'id': 'product'}, {'local': 'it-IT', 'fields': {'name': 'LeBron 17 Low "Glow in the Dark"', 'sku': 'CD5007-005', 'listingImage': [{'sys': {'type': 'Link', 'linkType': 'Asset', 'id': '1n7HNLzDCrSMwIVRTV76DS'}}, {'sys': {'type': 'Link', 'linkType': 'Asset', 'id': '4IozNCyTVmmveemPfyLq4u'}], {'sys': {'type': 'Link', 'linkType': 'Asset', 'id': '4kHEvV30gjqcuKlXouf8B'}}], {'sys': {'type': 'Link', 'linkType': 'Asset', 'id': '7b6VuPBpydc3HfrA5wf3'}], {'sys': {'type': 'Link', 'linkType': 'Asset', 'id': 'X1JpEENoPYQ76MgHiKOCE'}}], 'colorLabel': 'Spruce Aura/Black/Racer Blue', 'description': {'data': {}, 'content': [{'data': {}}, {'content': [{'data': {}}, {'marks': []}], 'value': 'A differenza della LeBron 16, nella LeBron 17 la versione Low torna a differire notevolmente dalla sorella a taglio alto. Decisamente pi  bassa a livello della caviglia, anche i materiali cambiano, con ritorni di mesh, suole e similpelle ad alternarsi in base alla colorazione. La scarpa risulta cos  molto alleggerita rispetto alla compagna pi  alta. Cambia anche il sistema di ammortizzazione: il React infatti si unisce alle gi  esistenti tecnologie, capitanate dal vistoso Air Max sul tallone. Il dettaglio esteticamente pi  particolare sta senza dubbio nei lacci: due linee differenti si inseguono, storte, su tutta la linguetta.', 'nodeType': 'text'}], 'nodeType': 'paragraph'}], 'nodeType': 'document'}, {'keyHighlights': [{'sys': {'type': 'Link', 'linkType': 'Entry', 'id': '503tmhBAgeR27LuGOvB4LC'}}, {'sys': {'type': 'Link', 'linkType': 'Entry', 'id': '6H1460dKYftTKdd35VC0c2'}}], {'sys': {'type': 'Link', 'linkType': 'Entry', 'id': 'Z1F0u0MTpn3K2R9VKRQWBDW'}], 'filterGender': [{'sys': {'type': 'Link', 'linkType': 'Entry', 'id': '3VmtZ4oksz2CmwmUGdy3nm'}}], 'filterColor': [{'sys': {'type': 'Link', 'linkType': 'Entry', 'id': '59GxmpeDIdYNgzdRDIBIP'}}], {'sys': {'type': 'Link', 'linkType': 'Entry', 'id': '24iQ0ysUq8tHe3ScIt7Ts1'}}], 'filterBrand': {'sys': {'type': 'Link', 'linkType': 'Entry', 'id': 'u13ieFXOYQ5kp3bcQtMQ'}}], 'filterCategory': {'sys': {'type': 'Link', 'linkType': 'Entry', 'id': '5DB856moqQM7Gy3vOCsre'}}], 'filterFamily': {'sys': {'type': 'Link', 'linkType': 'Entry', 'id': 'xIQLOmstptsXSLOTRXocr'}}], 'collection': None, 'outlet': False, 'restockable': False, 'release': '2020-09-15T09:00+02:00', 'listingsOrder': 26}}, {'sys': {'space': {'sys': {'type': 'Link', 'linkType': 'Space', 'id': '40i19ww9637w'}, 'id': 'xIQLOmstptsXSLOTRXocr', 'type': 'Entry', 'createdAt': '2019-12-17T09:22:20.347Z', 'updatedAt': '2019-12-17T09:22:20.347Z', 'environment': {'sys': {'id': 'master', 'type': 'Link', 'linkType': 'Environment'}}, 'revision': 1, 'contentType': {'sys': {'type': 'Link', 'linkType': 'ContentType', 'id': 'image/jpeg'}}, 'local': 'it-IT', 'fields': {'name': 'Performance'}, 'Asset': [{'sys': {'space': {'sys': {'type': 'Link', 'linkType': 'Space', 'id': '40i19ww9637w'}, 'id': '1n7HNLzDCrSMwIVRTV76DS', 'type': 'Asset', 'createdAt': '2020-09-14T08:38:55.088Z', 'updatedAt': '2020-09-14T08:38:55.088Z', 'environment': {'sys': {'id': 'master', 'type': 'Link', 'linkType': 'Environment'}}, 'revision': 1, 'local': 'it-IT', 'fields': {'title': 'lebron-17-low-glow-1', 'file': {'url': '//images.ctfassets.net/40i19ww9637w/1n7HNLzDCrSMwIVRTV76DS/bb59nd12dc5c772c31dfbbe7661b7db0f/lebron-17-low-glow-1.jpg', 'details': {'size': 134560, 'image': {'width': 1920, 'height': 1920}}, 'fileName': 'lebron-17-low-glow-1.jpg', 'contentType': 'image/jpeg'}}, {'sys': {'space': {'sys': {'type': 'Link', 'linkType': 'Space', 'id': '40i19ww9637w'}, 'id': '4IozNCyTVmmveemPfyLq4u', 'type': 'Asset', 'createdAt': '2020-09-14T08:38:56.142Z', 'updatedAt': '2020-09-14T08:38:56.142Z', 'environment': {'sys': {'id': 'master', 'type': 'Link', 'linkType': 'Environment'}}, 'revision': 1, 'local': 'it-IT', 'fields': {'title': 'lebron-17-low-glow-3', 'file': {'url': '//images.ctfassets.net/40i19ww9637w/4IozNCyTVmmveemPfyLq4u/07febdf2063bde43c540928d42717fc/lebron-17-low-glow-3.jpg', 'details': {'size': 123584, 'image': {'width': 1920, 'height': 1920}}, 'fileName': 'lebron-17-low-glow-3.jpg', 'contentType': 'image/jpeg'}}, {'sys': {'space': {'sys': {'type': 'Link', 'linkType': 'Space', 'id': '40i19ww9637w'}, 'id': '4kHEvV30gjqcuKlXouf8B', 'type': 'Asset', 'createdAt': '2020-09-14T08:38:57.036Z', 'updatedAt': '2020-09-14T08:38:57.036Z', 'environment': {'sys': {'id': 'master', 'type': 'Link', 'linkType': 'Environment'}}, 'revision': 1, 'local': 'it-IT', 'fields': {'title': 'lebron-17-low-glow-2', 'file': {'url': '//images.ctfassets.net/40i19ww9637w/4kHEvV30gjqcuKlXouf8B/96a3f4fc8d14454de17c3d98d7468b96/lebron-17-low-glow-2.jpg', 'details': {'size': 88578, 'image': {'width': 1920, 'height': 1920}}, 'fileName': 'lebron-17-low-glow-2.jpg', 'contentType': 'image/jpeg'}}, {'sys': {'space': {'sys': {'type': 'Link', 'linkType': 'Space', 'id': '40i19ww9637w'}, 'id': '7b6VuPBpydc3HfrA5wf3', 'type': 'Asset', 'createdAt': '2020-09-14T08:38:55.855Z', 'updatedAt': '2020-09-14T08:38:55.855Z', 'environment': {'sys': {'id': 'master', 'type': 'Link', 'linkType': 'Environment'}}, 'revision': 1, 'local': 'it-IT', 'fields': {'title': 'lebron-17-low-glow-4', 'file': {'url': '//images.ctfassets.net/40i19ww9637w/7b6VuPBpydc3HfrA5wf3/8c8595bdc56c7288e4dfa2b2354e8399/lebron-17-low-glow-4.jpg', 'details': {'size': 147389, 'image': {'width': 1920, 'height': 1920}}, 'fileName': 'lebron-17-low-glow-4.jpg', 'contentType': 'image/jpeg'}}, {'sys': {'space': {'sys': {'type': 'Link', 'linkType': 'Space', 'id': '40i19ww9637w'}, 'id': 'X1JpEENoPYQ76MgHiKOCE', 'type': 'Asset', 'createdAt': '2020-09-14T08:38:54.989Z', 'updatedAt': '2020-09-14T08:38:54.989Z', 'environment': {'sys': {'id': 'master', 'type': 'Link', 'linkType': 'Environment'}}, 'revision': 1, 'local': 'it-IT', 'fields': {'title': 'lebron-17-low-glow-5', 'file': {'url': '//images.ctfassets.net/40i19ww9637w/X1JpEENoPYQ76MgHiKOCE/567cb624136ca2e15c2abe7275dc938/lebron-17-low-glow-5.jpg', 'details': {'size': 214890, 'image': {'width': 1920, 'height': 1920}}, 'fileName': 'lebron-17-low-glow-5.jpg', 'contentType': 'image/jpeg'}}]]}}]
```

L'aggiunta al carrello avviene inserendo nel body di una richiesta POST all'indirizzo https://airness-2.commercelayer.io/api/line_items/ con i seguenti parametri

88

```

params = {
    'include': 'line_items,billing_address,shipping_address,shipments.shipment_line_items.line_item,
    'shipments.available_shipping_methods,shipments.shipping_method,available_payment_methods,payment_method,payment_source'
}

payload = '{"data":{"type":"orders","id":"","pid":"","attributes":{"shipping_address_same_as_billing":true},"relationships":{"billing_address":{"data":{"type":"addresses","id":"","ship_id":""}}}}}'

print(threadTime() + bcolors.WARNING + 'Submitting billing info..' + bcolors.ENDC)
logging.info('[AIRNESS] ' + threadTime() + 'Submitting billing info..')

while True:
    try:
        r = s.patch('https://airness-2.commercelayer.io/api/orders/' + pid, headers=headers, params=params, data=payload, proxies=proxy)
        checkShip = json.loads(r.text)

        if str(r.status_code) == '503':
            print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..')
            logging.info('[AIRNESS] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
            loadProxies()
            continue

```

Fig. 2.44 Istruzioni relative all'impostazione delle informazioni di fatturazione su Airness

```

params = {
    'include': 'shipment_line_items.line_item,available_shipping_methods,shipping_method'
}

payload = '{"data":{"type":"shipments","id":"","shipments_id":"","relationships":{"shipping_method":{"data":{"type":"shipping_methods","id":"","shipping_methods_id":""}}}}}'

print(threadTime() + bcolors.WARNING + 'Setting shipping info..' + bcolors.ENDC)
logging.info('[AIRNESS] ' + threadTime() + 'Setting shipping info..')

while True:
    try:
        r = s.patch('https://airness-2.commercelayer.io/api/shipments/' + shipments_id, headers = headers, params=params, data=payload, proxies=proxy)
        checkCourier = json.loads(r.text)

        if str(r.status_code) == '503':
            print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..')
            logging.info('[AIRNESS] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
            loadProxies()
            continue

```

Fig. 2.45 Istruzioni relative all'impostazione della corriere su Airness

```

params = {
    'include': 'line_items,billing_address,shipping_address,shipments.shipment_line_items.line_item,shipments.available_shipping_methods,
    'shipments.shipping_method,available_payment_methods,payment_method,payment_source'
}

payload = '{"data":{"type":"orders","id":"","pid":"","relationships":{"payment_method":{"data":{"type":"payment_methods","id":"","payment_method_id":""}}}}}'

print(threadTime() + bcolors.WARNING + 'Submitting payment info..' + bcolors.ENDC)
logging.info('[AIRNESS] ' + threadTime() + 'Submitting payment info..')

while True:
    try:
        r = s.patch('https://airness-2.commercelayer.io/api/orders/' + pid, headers = headers, params=params, data = payload, proxies=proxy)

        if str(r.status_code) == '503':
            print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..')
            logging.info('[AIRNESS] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
            loadProxies()
            continue

```

Fig. 2.46 Istruzioni relative all'impostazione delle informazioni di pagamento su Airness

Infine, il link per il pagamento è contenuto nella risposta alla richiesta POST all'indirizzo https://airness-2.commercelayer.io/api/paypal_payments utilizzando le seguente payload all'interno del body

```

firstURL = 'https://checkout.airness.it/'+pid+'/paypal'
secondURL = 'https://checkout.airness.it/'+pid

payload = '{"data":{"type":"paypal_payments","attributes":{"return_url":"' + firstURL + "','cancel_url":"' + secondURL + "'},
"relationships":{"order":{"data":{"type":"orders","id":"' + pid + "'}}}}}'

```

Fig. 2.47 Istruzioni relative a ottenere il link per il pagamento tramite PayPal su Airness

Esempio di link per il pagamento tramite PayPal è il seguente:
https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&token=EC-0TU61375GP0858405.

3.5.4 Modulo Shopify

Poiché il modulo Shopify è stato precedentemente descritto, nel seguente paragrafo verranno trattate le varie modalità sviluppate per questo modulo. A differenza degli altri moduli dove difficilmente è possibile visualizzare tutti i prodotti attualmente disponibili, nel caso di Shopify sono tutti presenti in un determinato file salvato lato server. Questo file, come detto in precedenza, è facilmente accessibile e contiene in particolare il nome del prodotto. Proprio per questo, una delle modalità di ricerca del prodotto è tramite keywords o parole chiavi.

```

{"name":"Colore","position":2,"values":["Nero"]}],{"id":"4655712927796","title":"Overreact Flyknit ISPA Nero","handle":"overreact-flyknit-isp-nero-2","body_html":"Utilizzando le ultime innovazioni del il nuovo progetto ISPA di Nike, il brand debutta con un nuovissimo modello chiamato Nike ISPA OverReact. Questa silhouette presenta una tomaia Flyknit intrecciata senza lacci con un'esclusiva clip in velcro bloccata utilizzata come sistema di allacciatura. Altri dettagli includono una linguetta a strappo sulla linguetta, dettagli riflettenti 3M in cima alla suola ed ammortizzata React ultra confortevole.", "published_at":"2020-09-10T11:39:38+02:00","created_at":"2020-09-09T01:07:56+02:00","updated_at":"2020-09-12T00:50:16+02:00","vendor":"Nike","product_type":"Sneakers","tags":["#3fa","2020-09-08T23:07:56Z","CD9664-002","color_Nero","filter-brand_Nike","filter-category_Sneakers","filter-color_Nero","filter-gender_unisex","filter-macro_Calzature","filter-macro_footwear","filter-price_\u20ac100 /\u20ac200","filter-shoesize_10","filter-shoesize_10.5","filter-shoesize_11","filter-shoesize_5","filter-shoesize_5.5","filter-shoesize_6","filter-shoesize_6.5","filter-shoesize_7","filter-shoesize_7.5","filter-shoesize_8","filter-shoesize_8.5","filter-shoesize_9","filter-shoesize_9.5","filter-type_Sneakers","gender_unisex","new","photo-yes","season_FW2020","sizegrid_SM2","style_CD9664-002"],"variants":[{"id":"32274030428212","title":"5 \\/ Nero","option1":"5","option2":"Nero","option3":null,"sku":"CD9664-#000#002#5","requires_shipping":true,"taxable":true,"featured_image":{"id":"14645919711284","product_id":"4655712927796","position":1,"created_at":"2020-09-09T17:24:32+02:00","updated_at":"2020-09-

```

Fig. 2.48 Informazioni relative al prodotto Overreact Flyknit ISPA nero disponibili sulla piattaforma Shopify

Se tutte le stringhe saranno presenti nel titolo del prodotto, quest'ultimo verrà aggiunto al carrello e successivamente verrà completato l'ordine. Un esempio è dato dal seguente prodotto selezionabile mediante parole chiavi "ISPA" e "Flyknit". L'algoritmo che procede alla selezione del prodotto mediante parole chiavi è il seguente:

```

for key in productDict.keys():
    if all(x in key for x in keywords):
        productName = key
        print(Fore.GREEN + "Product Found: " + productName)
        print(Style.RESET_ALL)
        productLink = productDict.get(key)
        # print(productLink)

        r = session.get(productLink, verify=False)

        variantDict = {}

        bs = soup(r.text, "html.parser")
        scripts = bs.findAll('script')
        jsonObj = None

        for s in scripts:
            if 'var meta' in s.text:
                script = s.text
                script = script.split('var meta = ')[1]
                script = script.split(';\\nfor (var attr in meta)')[0]

                jsonStr = script
                jsonObj = json.loads(jsonStr)

        for value in jsonObj['product']['variants']:
            variantSize = value['public_title']
            variantID = value['id']
            variantDict.update({variantSize: variantID})

```

Fig. 2.49 Istruzione relative alla selezione del prodotto mediante parole chiavi (keywords)

Un'altra modalità è rappresentata dal classico URL del prodotto nel caso in cui l'articolo desiderato sia già caricato sul sito e si possa accedere direttamente al sito. La scelta della modalità avviene direttamente tramite l'interfaccia CLI del software tramite una scelta con opzione.

Infine, tramite il file task.csv, è possibile scegliere la modalità di acquisto. Sono previste, infatti, due modalità:

- “MANUAL” in cui il bot provvederà a ricercare il prodotto, aggiungere la taglia desiderata e inviare tramite webhook un link per completare l'acquisto manualmente inserendo i propri dati di spedizione e di pagamento. Tale link individua univocamente una sessione di completamento dell'ordine. Di seguito viene riportato un esempio:

https://it.oneblockdown.it/7014842420/checkouts/42d28cfd8cd8eb59424fd4e6d82efad1?no_cookies_from_redirect=1

ONE BLOCK DOWN

Informazioni > Spedizione > Pagamento > Decidi

Check-out rapido

PayPal

OPPURE

Recapiti Hai già un account? Accedi

Email

Indirizzo di spedizione

Nome

Cognome

Codice Fiscale

Azienda (facoltativo)

Indirizzo

Interno (facoltativo)

Città

Paese / Regione
Italia

Provincia
Provincia

CAP

Telefono

1 Overreact Sandal ISPA Nero
4 / Nero €96,00

Gift card o codice sconto Applica

Subtotale €96,00

Spedizione ? Compila il modulo per stimare le spese di spedizione

Totale
Incluse imposte per
un ammontare di
€17,31 **€96,00**

Fig. 2.50 Pagina relativa alla sessione di ordine univoca

- “FAST” in questa modalità dovranno essere inseriti nel file profiles.json i dati per la spedizione e per il pagamento in modo tale che il bot possa completare in modo autonomo l’acquisto.

4. Tecnologie utilizzate

Nel seguente capitolo verranno descritti gli strumenti utilizzati per lo sviluppo del software, soffermandosi sulle motivazioni che mi hanno portato alla loro scelta.

4.1 Python

Python è un linguaggio di programmazione ad alto livello orientato ad oggetti e ideato da Guido van Rossum agli inizi degli anni Novanta. Tra gli obiettivi del linguaggio si distinguono dinamicità, semplicità e flessibilità. Queste peculiarità si traducono in un linguaggio in cui le variabili non sono tipizzate e in un uso dell'indentazione per la sintassi delle specifiche, piuttosto che le più comuni parentesi. In Python il controllo dei tipi è forte e viene eseguito a runtime: una variabile è un contenitore a cui viene associata una label che può essere correlata a contenitori di diverso tipo durante il suo ciclo vitale. Infine, Python utilizza un sistema di garbage collector per la liberazione e il recupero automatico della memoria di lavoro.

La mia personale esperienza a livello di linguaggi di programmazione si limitava ai linguaggi di Java e C. Tuttavia non è stato difficile apprendere Python, grazie soprattutto alla sua sintassi semplice. Il primo vantaggio che la scelta di questo linguaggio ha apportato al progetto è stata la sua forte integrazione con i servizi web, in ambito di sviluppo del software stesso. Python non è infatti stato utilizzato univocamente per gli algoritmi che si occupano di acquistare i prodotti, ma anche per la costruzione dell'intero sistema che gestisce l'autenticazione degli utenti lato server. Questo dimostra come il linguaggio sia estremamente completo direttamente dalla libreria standard, la quale fornisce numerosi moduli per lavorare con e-mail, pagine web, database e altro. Un secondo vantaggio portato dalla scelta di Python è la facilità con cui il linguaggio possa essere utilizzato su qualsiasi dispositivo, sia esso Windows, Linux/UNIX, Mac o mobile. Infatti, nonostante Python non sia un linguaggio

completamente interpretato, il codice sorgente non viene direttamente convertito in linguaggio macchina ma passa prima da una fase di precompilazione in bytecode, spesso utilizzata dopo la prima esecuzione del programma, evitando così di reinterpretare ogni volta la sorgente e migliorando le prestazioni. Nonostante le prestazioni non siano tra le migliori, come nel caso di linguaggi compilati statically typed (es. C), esse non influiscono sull'efficienza del bot. Infatti, il tempo di acquisto del prodotto è dovuto in gran parte dal tempo di risposta del server e non dall'effettiva velocità computazionale.

Di seguito vengono riportati e descritti i principali moduli non appartenenti alla libreria standard necessari allo sviluppo del software.

4.1.1 Requests ^[44]

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type":"User"... '
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

Fig. 4.1 Codice di esempio modulo Requests

Requests è un modulo Python che permette di effettuare richieste HTTP/1.1 in maniera estremamente facile e intuitiva. Come suggerisce la pagina di riferimento, non è necessario aggiungere manualmente una stringa query all'URL o codificare i dati trasmessi tramite POST. Una volta importato il modulo nel progetto sarà possibile effettuare richieste all'URL desiderato.

```
>>> r = requests.get('https://api.github.com/events')
```

Fig. 4.2 Codice di esempio richiesta GET eseguita utilizzando il modulo Requests

Si ottiene in questo modo un oggetto risposta (variabile `r`), dal quale è possibile avere una serie di informazioni legate alla risposta del server, alla richiesta effettuata, sullo status code, sull'URL restituito, sull'header e sul contenuto del body. In questo modo sarà possibile verificare se la richiesta effettuata sia andata a buon fine (status code appartenente al gruppo che assume valore due come prima cifra), se restituisca un link di pagamento e se possa estrapolare le informazioni di interesse dalla risposta.

In maniera analoga alla richiesta di GET è possibile effettuare richieste POST.

```
>>> r = requests.post('https://httpbin.org/post', data = {'key':'value'})
```

Fig. 4.3 Codice di esempio richiesta POST eseguita utilizzando il modulo Requests

In questo modo la richiesta POST contiene nel proprio body le informazioni passate come un dizionario di stringhe.

“I dizionari (dict) sono un tipo built-in, mutabile e non ordinato che contiene elementi (items) formati da una chiave (key) e un valore (value). Una volta che il dizionario è creato e valorizzato con un insieme di coppie <chiave, valore>, si può usare la chiave (che deve essere univoca) per ottenere il valore corrispondente. [45]”

Infine, il modulo mette a disposizione la possibilità di utilizzare headers personalizzabili semplicemente passando un dizionario come parametro di header nella funzione.

```
>>> url = 'https://api.github.com/some/endpoint'
>>> headers = {'user-agent': 'my-app/0.0.1'}

>>> r = requests.get(url, headers=headers)
```

Fig. 4.4 Codice di esempio richiesta GET con header personalizzato eseguita utilizzando il modulo Requests

Questa possibilità risulta indispensabile per lo sviluppo del software in quanto, come affermato in precedenza, la maggior parte dei sistemi di protezione utilizzati dagli e-commerce sono facilmente eludibili utilizzando un header correttamente formattato.

4.1.2 BeautifulSoup [46]

Beautiful Soup rappresenta, insieme a Request, uno dei moduli fondamentali per lo sviluppo del progetto. Infatti esso rende semplice estrarre informazioni da file HTML e XML. Utilizzando una serie di parser è in grado di fornire diversi metodi per navigare, ricercare e modificare l'albero della pagina. Una volta importato il modulo, sarà possibile creare un oggetto “soup” a partire dalla risposta ad una richiesta. Tale oggetto include le informazioni riguardanti la struttura e il contenuto del file ottenuto dal parser che si desidera utilizzare.

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
```

Fig. 4.5 Codice di esempio modulo BeautifulSoup

Successivamente sarà possibile estrarre gli elementi ricercati.

```
soup = BeautifulSoup(r.text, 'html.parser')
productIMG = 'https://www.susi.it' + str(soup.find('img', {'class': 'primary-image'}))['src']
productTitle = ""
productTitle = soup.find('h2', {'class': 'product-name'}).text
if(productTitle == ""):
    productTitle = soup.find('div', {'class': 'label'}).text.split(':')[1]

sizeDict = {}
hrefs = soup.find_all('a', {'class': 'swatchanchor'})
i = 0
for href in hrefs:
    if i == 0:
        i = i + 1
        continue

    link = href['href'] + '&Quantity=1&format=ajax&productlistid=undefined'

    r = session.get(link, headers=headersXML)

    if str(r.status_code) == '503':
        print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..' + bcolors.ENDC)
        logging.info('[SUSI] ' + threadTime() + '[' + str(r.status_code) + ']' + '[ACCESS DENIED] Proxy banned, retrying with a new one..')
        loadProxies()
        continue

    soup = BeautifulSoup(r.text, 'html.parser')
    productID = soup.find('span', {'itemprop': 'productID'})
    sizeDict.update({productID.text: str(href.text).replace('\n', '')})
```

Fig. 4.6 Codice di esempio utilizzo modulo BeautifulSoup al fine di estrarre informazioni dal codice HTML

Nell'esempio riportato, viene creato un oggetto soup a partire dalla risposta (r.text) ottenuta dalla richiesta all'URL del prodotto. Dall'oggetto soup vengono dunque estrapolate:

- Immagine del prodotto ricavato dal valore di “src” cercando fra tutti gli elementi di tag “img” appartenenti alla classe “primary-img”.

```

```

Fig. 4.7 Immagine del prodotto ricavato dal valore di “src”

- Titolo del prodotto cercando il contenuto dell’elemento di tag “h2” di classe “product-name”

```
<h1 class="product-brand" itemprop="brand">ADIDAS</h1>
<h2 class="product-name" itemprop="name">SCARPA FREE HIKER PARLEY</h2>
```

Fig. 4.8 Titolo del prodotto cercando il contenuto dell’elemento di tag “h2” di classe “product-name”

- Informazioni sulle taglie trovando tutti i link di tag “a” di classe “swatchanchor”

```
<li class="selectable">
  <a class="swatchanchor"
    href="https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/Product-Variation?pid=ADIDAS%2e-EF0347&dwvar_ADIDAS%2e-EF0347_size=290&dwvar_ADIDAS%2e-EF0347_color=86797"
    title="Seleziona Taglia: 6.5"
    data-lgimg="{&quot;url&quot;:&quot;/on/demandware.static/-/Sites-susi-catalog-master/default/dwcd09fec2/images/large/2000215675770/200021567577001.jpg&quot;,&quot;title&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK&quot;,&quot;alt&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK, large&quot;,&quot;hires&quot;:&quot;,&quot;}">
    6.5
  </a>
</li>

<li class="selectable">
  <a class="swatchanchor"
    href="https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/Product-Variation?pid=ADIDAS%2e-EF0347&dwvar_ADIDAS%2e-EF0347_size=291&dwvar_ADIDAS%2e-EF0347_color=86797"
    title="Seleziona Taglia: 7"
    data-lgimg="{&quot;url&quot;:&quot;/on/demandware.static/-/Sites-susi-catalog-master/default/dwcd09fec2/images/large/2000215675770/200021567577001.jpg&quot;,&quot;title&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK&quot;,&quot;alt&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK, large&quot;,&quot;hires&quot;:&quot;,&quot;}">
    7
  </a>
</li>

<li class="selectable">
  <a class="swatchanchor"
    href="https://www.susi.it/on/demandware.store/Sites-susi-Site/it_IT/Product-Variation?pid=ADIDAS%2e-EF0347&dwvar_ADIDAS%2e-EF0347_size=292&dwvar_ADIDAS%2e-EF0347_color=86797"
    title="Seleziona Taglia: 7.5"
    data-lgimg="{&quot;url&quot;:&quot;/on/demandware.static/-/Sites-susi-catalog-master/default/dwcd09fec2/images/large/2000215675770/200021567577001.jpg&quot;,&quot;title&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK&quot;,&quot;alt&quot;:&quot;SCARPA FREE HIKER PARLEY, COREBLACK, large&quot;,&quot;hires&quot;:&quot;,&quot;}">
    7.5
  </a>
</li>
```

Fig. 4.9 Informazioni sulle taglie trovando tutti i link di tag “a” di classe “swatchanchor”

(URL di riferimento: https://www.susi.it/it-IT/uomo/scarpe/sneaker/scarpa-free-hiker-parley/ADIDAS.-EF0347.html?dwvar_ADIDAS.-EF0347_color=86797&cgid=803#start=2)

4.1.3 Cloudscraper [47]

Cloudscraper è un modulo Python utilizzato per eludere la pagina anti-bot di CloudFlare (conosciuta anche come “I’m Under Attack Mode”, or IUAM) basato su Requests. Come suggerito dalla sito web del modulo, quest’ultimo è utile per eseguire web scraping di siti protetti con CloudFlare. Al momento il sistema di protezione controlla se chi effettua la richiesta supporta Javascript. Per questo motivo, cloudscraper richiede un interprete Javascript per risolvere le sfide di verifica proposte. Questo permette all’algoritmo di impersonare facilmente un browser regolare senza eseguire parsing esplicito del codice Javascript di CloudFlare.

Il modulo si integra perfettamente con Requests, in modo tale che lo sviluppatore possa eseguire le richieste in modalità analoga. Infatti, una volta creato l’oggetto cloudscraper

```
s = cloudscraper.create_scraper(interpreter='js2py')
```

esso potrà essere utilizzato per effettuare richieste GET o POST con i parametri descritti in precedenza.

```
r = s.get(url, headers=headersBase, proxies=proxy)

if str(r.status_code) == '503':
    print(threadTime() + bcolors.FAIL + '[ACCESS DENIED] Proxy banned, retrying with a new one..' + bcolors.ENDC)
    logging.info('[GROSBASKET] ' + threadTime() + '[' + str(r.status_code) + '] [ACCESS DENIED] Proxy banned, retrying with a new one..')
    loadProxies()
    continue
```

Fig. 4.10 Istruzioni di esempio dell’uso del modulo Cloudscraper

4.1.4 Selenium [48]

Selenium Python mette a disposizione una serie di semplici API per sviluppare funzioni a fini di test usando il Web Driver Selenium. Il Web Driver opera su un browser in modo nativo, come farebbe un utente, localmente o su una macchina remota utilizzando il server Selenium e garantendo una perfetta automazione del browser. Una volta inserito il modulo all’interno del proprio codice, sarà necessario disporre del Driver Chrome all’interno del path di lancio dell’algoritmo.

```

driver = webdriver.Chrome('data/chromedriver.exe')
driver.get('https://paypal.com')
driver.delete_all_cookies()
for c in s.cookies:
    z = {'name':c.name, 'value':c.value, 'path':c.path}
    driver.add_cookie(z)

driver.get(pay.url)

```

Fig. 4.10 Algoritmo utilizzato per inserire i cookie di sessioni all'interno del Web Driver Selenium

In questo modo il browser aperto potrà aprire la pagina riferita al pagamento, eliminare i cookie presenti e successivamente aggiungere quelli relativi alla sessione di acquisto del prodotto, prelevati direttamente sfruttando le funzionalità offerte da Request.

4.1.5 Threading ^[49]

Poiché lo scopo del bot è quello di finalizzare il numero massimo di acquisti nel minor tempo possibile, oltrepassando i limiti di un utente manuale, risulta necessario adoperare una programmazione concorrente. *“Un'applicazione che sfrutta questo paradigma si dice multithread. I thread non sono altro che sotto-processi eseguiti generalmente in parallelo, e generati da un processo padre. Quando uno script Python viene eseguito, abbiamo infatti la possibilità di creare uno o più thread, che possono collaborare per il raggiungimento di uno scopo comune, eventualmente condividendo le stesse risorse computazionali, nonché gli stessi dati. I processori dei computer più recenti sono generalmente multicore, offrendo quindi la possibilità di eseguire più operazioni parallele, sfruttando al meglio le risorse computazionali del calcolatore.* ^{[50]”}

Come descritto nel capitolo precedente, l'utente che utilizza il bot imposta il prodotto e la taglia d'interesse utilizzando il file task.csv relativo al sito dedicato al rilascio.

task.csv - Ron's Editor - licensed to Evaluator

File Edit View Row Column Cell Filter Tools Help

Main Document

task.csv

	URL	SIZE
01	https://www.7hillsroma.com/it/vans-old-skool-pro-black-black.html	RANDOM
02	https://www.7hillsroma.com/it/vans-old-skool-pro-black-black.html	4-7
03	https://www.7hillsroma.com/it/footwear/nike-sb/nike-sb-blazer-low-gt-black-sail.html	7
04	https://www.7hillsroma.com/it/footwear/ripndip/ripndip-nerm-high-black-8613.html	8
05	https://www.7hillsroma.com/it/footwear/adidas-skateboarding/adidas-skateboarding-superstar-adv-black-white-gold.html	RANDOM
06	https://www.7hillsroma.com/it/footwear/nike-sb/nike-sb-blazer-low-gt-black-sail.html	RANDOM
07	https://www.7hillsroma.com/it/vans-old-skool-pro-black-black.html	RANDOM
08	https://www.7hillsroma.com/it/vans-old-skool-pro-black-black.html	RANDOM
09	https://www.7hillsroma.com/it/footwear/ripndip/ripndip-nerm-high-black-8613.html	RANDOM
10	https://www.7hillsroma.com/it/footwear/ripndip/ripndip-nerm-high-black-8613.html	RANDOM
11	https://www.7hillsroma.com/it/footwear/adidas-skateboarding/adidas-skateboarding-superstar-adv-black-white-gold.html	4-10
12	https://www.7hillsroma.com/it/footwear/ripndip/ripndip-nerm-high-black-8613.html	9
13	https://www.7hillsroma.com/it/footwear/nike-sb/nike-sb-blazer-low-gt-black-sail.html	7
14	https://www.7hillsroma.com/it/footwear/adidas-skateboarding/adidas-skateboarding-superstar-adv-black-white-gold.html	6-12
15	https://www.7hillsroma.com/it/footwear/ripndip/ripndip-nerm-high-black-8613.html	RANDOM
16	https://www.7hillsroma.com/it/footwear/nike-sb/nike-sb-blazer-low-gt-black-sail.html	RANDOM

Ron's Editor Lite 100% free
After evaluation commercial entities are required to have a Pro license.

Rons Data Cleanser
Complementing Rons CSV Editor, an application that automates cleaning, processing and converting large CSV and data files.

Go Pro
Try Pro Free

Column Count | 2 Row Count | 16 Status Source: task.csv Column - Row - Selected Cells - Text Length -

Fig. 4.11 Esempio di file task.csv

Per ogni riga del file, verrà generato un thread indipendente che svolgerà l'acquisto utilizzando i dati del prodotto e di spedizione dell'utente relativi alla riga di riferimento.

```
with open('susi/task.csv', 'r') as csv_file:
    csv_key = csv.DictReader(csv_file)

    for line in csv_key:
        try:
            email = line['EMAIL']
            name = line['FIRST NAME']
            if('RANDOM' in email):
                if('RANDOM' in line['FIRST NAME']):
                    name, email = utility.getCatcallEmail(line['LAST NAME'], email.replace('RANDOM', ''))
            phone_number = line['PHONE']

            if('RANDOM' in phone_number):
                phone_number = utility.getRandomNumber()

            accountDict = {
                'email': email,
                'name': name,
                'last name': line['LAST NAME'],
                'address': line['ADDRESS'],
                'address_2': line['ADDRESS 2'],
                'city': line['CITY'],
                'zipcode': line['ZIPCODE'],
                'phone number': phone_number,
                'province': line['PROVINCE'],
                'country': line['COUNTRY']
            }

        except Exception as e:
            print(threadTime() + bcolors.FAIL + 'Failed loading task, please check csv file!' + bcolors.ENDC)
            logging.info('[SUSI] ' + threadTime() + '[' + str(e) + '] Failed loading task, please check csv file!')
            input('Press ENTER to exit.')
            sys.exit(-1)

    t = threading.Thread(target=main, args=(line['URL'], line['SIZE'], accountDict))
    t.start()
```

Fig. 4.12 Algoritmo che preleva le informazioni dal file task.csv e avvia i thread

Viene dunque aperto il file in formato csv utilizzando un apposito modulo. Tramite un ciclo for vengono passate tutte le righe e prelevate le informazioni relative al prodotto creando così un dizionario di stringhe contenente tutti i parametri relativi alla spedizione. Per ogni ciclo, viene realizzato un oggetto Thread avente come target la funzione main con parametri URL del prodotto, taglia desiderata e il dizionario precedente creato. Infine viene avviato il thread utilizzando la funzione start(). Ogni thread utilizzerà un proxy diverso (prelevato dal file proxies.txt) in modo tale che le richieste appaiano al sito come provenienti da diversi indirizzi logici ed evitando dunque di essere rintracciati dai sistemi anti-bot.

4.1.6 Discord Webhook ^[51]

Come descritto nel precedente capitolo, l'utente sarà notificato dalla piattaforma Discord dopo il completamento dell'ordine. Questo è reso possibile grazie al modulo discord-webhook. Una volta importato il modulo, sarà possibile creare un oggetto webhook contenente le informazioni inerenti al prodotto e caratterizzato dall'URL del canale dove verrà inviata la notifica.

```
from discord_webhook import DiscordWebhook, DiscordEmbed

webhook = DiscordWebhook(url='your webhook url')

# create embed object for webhook
embed = DiscordEmbed(title='Your Title', description='Lorem ipsum dolor sit', color=242424)

# add embed object to webhook
webhook.add_embed(embed)

response = webhook.execute()
```

Fig. 4.13 Codice di esempio di utilizzo del modulo DiscordWebhook

Nel software sviluppato viene utilizzata un'unica funzione per tutti i siti basata sulle peculiarità che li accomunano.

```

def sendWebhook(site, pUrl, email, ProductName, size, cUrl, img, payment, time):

    embed = DiscordEmbed(title='Successfully checked out!', color=0xff00dd)

    embed.add_embed_field(name='Website', value=site, inline=False)
    embed.add_embed_field(name='Product', value='['+ProductName+']('+pUrl+')', inline=False)
    embed.add_embed_field(name='Size', value=size, inline=False)

    if '@' in email:
        embed.add_embed_field(name='Account', value='['+email+']', inline=False)

    if(payment == 'Cash on delivery'):
        embed.add_embed_field(name='Order Number', value='['+cUrl+']')
    else:
        embed.add_embed_field(name='Checkout Link', value='[Click here]('+cUrl+')')

    embed.add_embed_field(name='Payment', value=payment, inline=False)

    final_time = str(time).split(':')[2]+'s'

    tmp = list(final_time)

    if tmp[0] == '0':
        tmp[0] = ''

    final_time = ''.join(tmp)

    embed.add_embed_field(name='Time', value=str(final_time), inline=False)

    embed.set_thumbnail(url=img)
    embed.set_footer(icon_url = criss_logo, text='CrissAIO • '+str(date.today()))

    webhook = DiscordWebhook(url = webhookLink)

    webhook.add_embed(embed)

    try:
        webhook.execute()
    except Exception as e:
        print/utility.threadTime() + bcolors.FAIL + ' Failed to send webhook, please check debug file for payment link!' + bcolors.ENDC
        logging.info('[WEBHOOK] ' + utility.threadTime() + payment)
        logging.info('[WEBHOOK] ' + utility.threadTime() + '[' + str(e) + '] Failed to send webhook, please check debug file for payment link!')
        pass

```

Fig. 4.14 Istruzioni utilizzate per inviare la notifica tramite DiscordWebhook in seguito al completamento dell'ordine

4.2 Google Chrome e Fiddler

Ogni acquisto online di un prodotto è determinato da una serie di operazioni che si restringono a semplici richieste GET o POST effettuate al sito in cui è rilasciato l'articolo stesso. Oltre alle richieste, ogni articolo è caratterizzato da una pagina HTML in cui è presentato. Al fine di estrapolare le corrette richieste e informazioni sono stati simulati degli acquisti per gli acquisti sviluppati. Questo è stato possibile utilizzando due software: Google Chrome e Fiddler. Tramite Google Chrome è stato analizzato il codice HTML delle pagine grazie alla funzionalità base di visualizzazione del codice sorgente. Fiddler ha realizzato invece un proxy il cui è obiettivo era quello di eseguire il debug di qualsiasi problema relativo al traffico HTTP tra il computer e Internet.

L'interfaccia è stata conseguentemente suddivisa in una serie di pagine che permettevano di accedere alle statistiche delle pagine che si visitano o di vedere, in diversi formati, il contenuto di ogni richiesta e la corrispondente risposta. Di seguito viene effettuata una simulazione di acquisto sul sito Susi in modo da prelevare le richieste necessarie per sviluppare l'algoritmo.

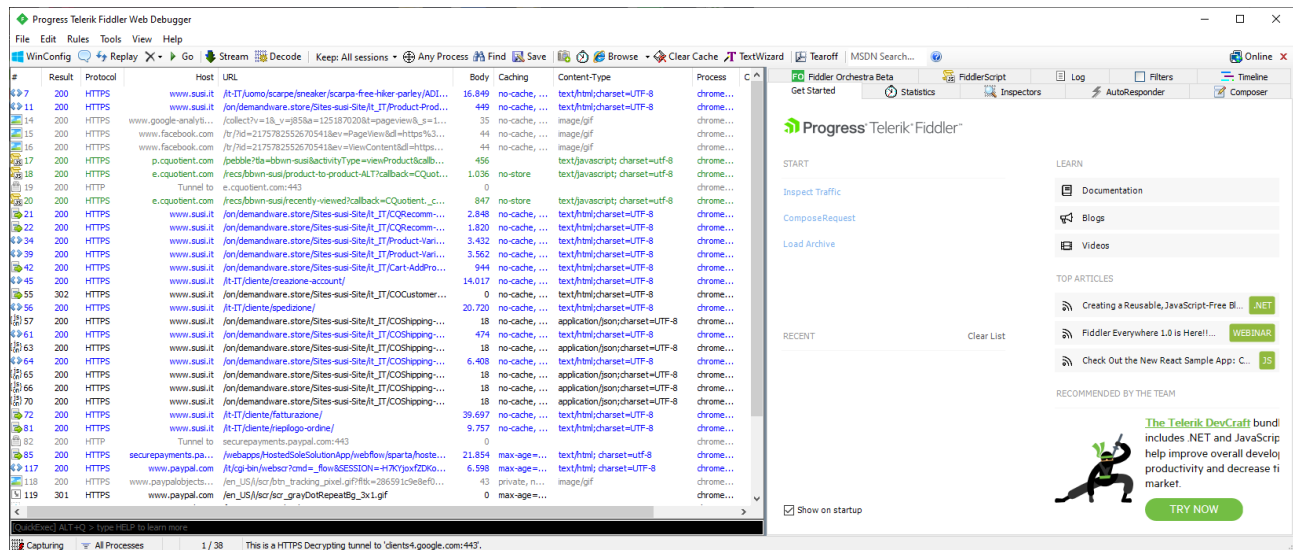


Fig. 4.15 Schermata di registrazione delle richieste dell'acquisto di un prodotto su Susi

4.3 Javascript

Javascript è un linguaggio di programmazione orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client per la creazione di effetti dinamici e interattivi. Il tutto è reso possibile grazie alle funzioni di script innescate in vari modi dall'utente sulla pagina web in uso. L'uso di tale linguaggio si limita alla costruzione dell'estensione per trasferire i cookie dalla sessione del bot alla propria. L'operazione rende l'intera esperienza dell'utente più veloce e meno dispendiosa in termini di risorse hardware. La scelta del linguaggio è tuttavia vincolata dalle specifiche richieste per lo sviluppo di una estensione Chrome. Infatti un'estensione comprende un file manifest.json in cui vengono definite tutte le caratteristiche dell'estensione, un file Javascript che serve da punto di accesso alla logica operativa e infine due file opzionali style.css che contengono eventuali stili e

immagine.png, nonché icone dell'estensione. Nel manifest è dunque possibile segnalare a Chrome cosa farà la nostra applicazione e di quali file o librerie avrà bisogno per funzionare.

```
{
  // È obbligatorio definire nome e versione dell'applicazione
  // il manifest che utilizziamo è quello della versione 2
  "name": "Estensione HTML.it",
  "version": "2.0",
  "manifest_version": 2,
  // Opzionale, indichiamo descrizione, formato e nome dell'immagine
  "description": "Esperimento su HTML.it",
  "icons": { "16": "immagine.png" },
  // Serve per dichiarare quali moduli o API di chrome utilizzeremo
  // nell'applicazione. Per questo esempio abbiamo indicato che
  // utilizzeremo i moduli chrome.tabs o chrome.windows
  "permissions": [
    "tabs"
  ],
  // Ecco la parte in cui definiamo i nostri script per i contenuti
  // e indichiamo a quali url applicare la nostra estensione
  "content_scripts":
  [
    {
      "matches": ["https://www.html.it/*"],
      "js": ["jquery.js", "azioni.js"],
      "css": ["stili.css"]
    }
  ]
}
```

Fig. 4.16 Esempio di manifesto per la creazione di una estensione Chrome

La logica applicativa dell'estensione è estremamente semplice. Essa riconosce l'attuale URL e si comporterà in modo analogo al Web Driver Selenium. Aprirà la pagina corrispondente al pagamento, sia esso PayPal o la pagina di verifica della carta di credito, cancellerà i cookie precedentemente presenti, aggiungerà i cookie passati direttamente come parametro nella query URL e infine reindirizzerà la pagina a quella relativa al pagamento.

4.4 Performance

Le performance degli algoritmi vengono misurate direttamente all'interno del codice. Questo è reso possibile dall'utilizzo di un timer perfetto che viene fatto partire dal momento in cui l'oggetto viene aggiunto al carrello e si conclude una volta disponibile il link per effettuare il pagamento. La partenza avviene in quel preciso istante poiché l'attività di login o di scraping della pagina può essere effettuata prima che il prodotto sia effettivamente rilasciato. Di conseguenza l'intervallo di tempo precedente all'aggiunta al carrello dell'articolo non influisce sull'effettiva efficienza dell'algoritmo.

```
start_time = datetime.now()
print(utility.threadTime('') + utility.bcolors.WARNING + 'Adding to cart size [' + sizeID + ']' + utility.bcolors.ENDC)
logging.info('[AIRNESS] ' + utility.threadTime('') + 'Adding to cart size [' + sizeID + ']')
addToCart(session, pid, variant, carted, checkout)

if(couponCode != ''):
    addCoupon(session, pid, couponCode)

setCustomerEmail(session, pid, accountDict['email'])
ship_id = setCustomerAddress(session, accountDict)

payment_method_id, shipments_id, shipping_methods_id = setBillingInfo(session, pid, ship_id)
setCourier(session, shipments_id, shipping_methods_id)

setPayment(session, pid, payment_method_id)
checkout_link = completeCheckout(session, pid, carted, checkout)

final_time = datetime.now() - start_time
```

Fig. 4.17 Istruzioni per le quali viene calcolato il tempo di completamento dell'ordine

Inoltre, è importante sottolineare quanto poco sia influente l'efficienza del codice. Infatti, il ritardo maggiore viene introdotto dalle richieste di GET o di POST e non da come sia ottimizzato il codice. Tale ritardo deriva da una parte dalla larghezza di banda a disposizione sul dispositivo utilizzato, dall'altra da eventuali proxy, se utilizzati, o dal traffico attuale sul server del sito su cui si desidera effettuare l'acquisto. I test riportati successivamente sono stati effettuati senza l'utilizzo di proxy. In caso fosse possibile, i risultati ottenuti saranno confrontati con le performance di altri sneaker bot attualmente a disposizione.

Le performance del modulo Shopify non saranno trattate, in quanto il tempo di completamento dell'ordine è strettamente vincolato dal tempo con cui viene risolta manualmente la sfida del captcha Google. Poiché il tempo di risoluzione spesso dipende anche dalla difficoltà stessa della sfida si è deciso di limitare i test di performance alle restanti piattaforme supportate.

PIATTAFORMA	SITO	URL SITO	MODE	URL PRODOTTO	PERFORMANCE BOT SVILUPPATO	BOT CONFRONTO	PERFORMANCE BOT CONFRONTO
DENANDVARE	SUSI	https://www.susi.it/	-	https://www.susi.it/it/omono/scarpe/anelker/scarpa-super.com/ADDONS-EE0208.html?new_ADDONS-EE0208_color=60778&gid=803	16,934155	-	-
DENANDVARE	SUSI	https://www.susi.it/	-	https://www.susi.it/it-1T/omono/scarpe/anelker/scarpa-free-like-pink/ADDONS-EE0347.html?new_ADDONS-EE0347_color=80797&gid=803	17,7915635	-	-
DENANDVARE	AVILAB	https://www.av-lab.com/	-	https://www.av-lab.com/demal/exclusive/flu-disruptor-low-AW_15991.P12_5046382_006.html	2,1388086	Okie	2,265
DENANDVARE	AVILAB	https://www.av-lab.com/	-	https://www.av-lab.com/uoed/scarpa-AW_2210808A_8011399_08.html	1,762,4275	Okie	3,145
MAGENTO	ALLINE	https://www.allinestore.com/	PAYPAL	https://www.allinestore.com/german/adidas-y-3-hocker-it-white-cream-95912-129269.html	5,9556025	Opelclio	3,6538
MAGENTO	ALLINE	https://www.allinestore.com/	CREDIT CARD	https://www.allinestore.com/default/vcp-sds-see-white-keps-black-ctrl012640-130089.html	11,4960345	Opelclio	8,19435
MAGENTO	GROSBASKET	https://www.grobsbasket.com/	-	https://www.grobsbasket.com/it/air-jordan-1-jester-xx-low-laced-wmns-black-white	4,3165825	Opelclio	2,28435
MAGENTO	GROSBASKET	https://www.grobsbasket.com/	-	https://www.grobsbasket.com/it/nike-sd-ivy-5-vill-smoke-grey	4,1515655	Opelclio	3,94325
MAGENTO	THILLS	https://www.thillsroma.com/	-	https://www.thillsroma.com/it/vars-saddle-sid-pro-black-white.html	19,0627035	-	-
MAGENTO	THILLS	https://www.thillsroma.com/	-	https://www.thillsroma.com/it/footwear/vans-skate-vans-er-pro-black-white-gun.html	19,0514305	-	-
MAGENTO	HOLYPOSTORE	https://www.holypostore.com/	SAFE	https://www.holypostore.com/it/footwear-umono/nike-air-zoom-splidon-cage-2/10674	11,2102145	ThunderIO	9,105
MAGENTO	HOLYPOSTORE	https://www.holypostore.com/	FAST	https://www.holypostore.com/it/footwear-umono/nike-air-zoom-splidon-cage-2/10674	5,2943215	ThunderIO	8,725
MAGENTO	SOTF	https://www.soft.com/	-	https://www.soft.com/it/superga-x-soft/omono/anelker/s/anelker-super-ga-x-soft-2941-ripectuene-57111cw.html	3,4795555	-	-
MAGENTO	SOTF	https://www.soft.com/	-	https://www.soft.com/it/off-white-umono/sneakers/sneakers-super-ga-x-soft-2941-ripectuene-57111cw.html	5,8127345	-	-
MAGENTO	ASPAHIT GOLD	https://www.aspahitgold.com/	CART HOLD	https://www.aspahitgold.com/en/product/nike-wmns-air-jordan-1-low-white-university-red-white/	0,8907005	ThunderIO	1,145
MAGENTO	ASPAHIT GOLD	https://www.aspahitgold.com/	CREDIT CARD	https://www.aspahitgold.com/en/product/nike-wmns-air-jordan-1-low-white-university-red-white/	21,1781805	ThunderIO	17,745
MAGENTO	CORNESTREET	https://www.cornestreet.it/	-	https://www.cornestreet.it/nike-air-jordan-4-retro-neon-air-max.html	5,385,454	ThunderIO	4,375
MAGENTO	CORNESTREET	https://www.cornestreet.it/	-	https://www.cornestreet.it/nike-air-force-1-lyt-ga-white-white-black.html	6,084,906	ThunderIO	4,415
WOOCOMMERCE	AIRNESS	https://airness.it/	-	https://airness.it/it/airson-1-low-glow-in-the-dark	3,0689155	ThunderIO	2,265
WOOCOMMERCE	AIRNESS	https://airness.it/	-	https://airness.it/it/syberid-2-nas	3,3161755	ThunderIO	2,175
WOOCOMMERCE	WORKING CLASS HEROES	https://www.workingclassheroes.co.uk/	PAYPAL	https://www.workingclassheroes.co.uk/footwear/trainers/nike-sh-px-return-premium-shoes-medium-black-sail_207758	1,6108465	ThunderIO	2,815
WOOCOMMERCE	WORKING CLASS HEROES	https://www.workingclassheroes.co.uk/	CREDIT CARD	https://www.workingclassheroes.co.uk/footwear/trainers/nike-sh-px-return-premium-shoes-medium-black-sail_207758	6,5176645	ThunderIO	9,135

Fig. 4.18 Tabella raffigurante le performance del software sviluppato sui diversi siti supportati

Conclusione

La capillare diffusione di Internet e la forte influenza dei social network hanno portato il fenomeno dello streetwear e delle sneakers a un livello di interesse che mai sarebbe stato immaginabile. La passione per oggettistica limitata si è evoluta nel corso del tempo: il tutto è nato dal senso di appartenenza e amore verso il brand, trasformandosi in necessità e ossessione per l'ostentazione, portando così il soggetto in questione a voler mostrare l'oggetto senza più alcun legame sentimentale con esso. Nonostante questi comportamenti possano sembrare un sintomo passeggero dettato dalla moda, Antonio Cristaudo, responsabile market developing di Pitti Immagine, sostiene "lo streetwear, inteso come fenomeno culturale e non come tendenza, è certamente destinato a durare". Di conseguenza, esisterà sempre un gruppo di persone disposte a tutto per ottenere il proprio "Sacro Graal". Ad assecondare questo circolo vizioso ci penseranno inoltre le case produttrici di sneakers, proponendo prodotti sempre più esclusivi e limitati, alimentando la "necessità" di ottenerli e di conseguenza l'inevitabile proliferazione dei bot a disposizione dell'utente che desidera ardentemente l'articolo. Nonostante possa sembrare scorretto nei confronti degli utenti che non posseggono tali software, la pratica è tuttavia utilizzata dalla maggior parte dei membri della community. "Hate the game, not the player". Questa celebre frase rappresenta al meglio il modo in cui la competizione, all'interno di questo mondo, abbia portato i propri membri a doversi adattare all'interno del mutevole sneakers game. Se in guerra vince chi ha l'arma migliore, in questo caso compra chi ha il bot più veloce.

Queste considerazioni vengono utilizzate per sottolineare come, nonostante possa sembrare estraneo ad occhi non appartenenti a questo mondo, gli sneaker bots rimarranno uno strumento necessario e largamente utilizzato per i prossimi decenni. L'intento del software sviluppato è proprio quello di evolversi insieme alle nuove piattaforme e-commerce e protezioni anti-bot in modo da fornire agli utilizzatori lo strumento più adeguato ad acquistare edizioni limitate.

Attualmente l'applicazione conta ben 12 siti supportati sulle 4 piattaforme trattate in precedenza e 35 utenti. Tali utenti sono risultati fondamentali al fine di riscontrare eventuali bug presenti negli algoritmi e per migliorare l'efficienza del sistema. L'applicazione viene proposta con una interfaccia semplice e reattiva in modo da favorirne l'utilizzo anche ai meno pratici del settore. Questa caratteristica, unita al supporto di siti poco conosciuti, permette agli utenti di acquistare molteplici paia ad ogni rilascio garantendo loro acquisti costanti. Il risultato deriva anche da performance perfettamente in linea con l'attuale concorrenza come mostrato nella tabella delle prestazioni precedentemente descritta (**Fig. 4.18**). L'intervallo di tempo che lo sneaker bot sviluppato impiega per acquistare prodotti sulle diverse piattaforme è pressoché lo stesso che si decida di lanciare una sola task o decine di esse, grazie ad un utilizzo corretto del modulo Threading. In questo modo è possibile aumentare notevolmente le proprie probabilità di acquisto senza alcun compromesso, lasciando la decisione all'utente in base alle proprie necessità.

Di seguito viene riportato uno dei migliori risultati ottenuti durante il periodo di beta.

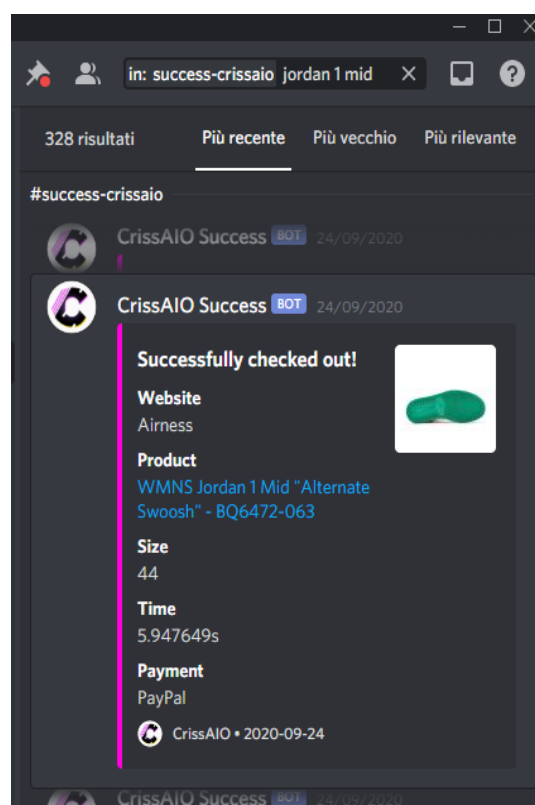


Fig. 5.1 Numero di acquisti completati con successo per il modello Jordan 1 Mid sulle diverse piattaforme

L'immagine mostra il numero di acquisti completati con successo sui diversi moduli supportati per il modello Jordan 1 Mid, uno dei prodotti più desiderati di quest'anno. I risultati sono stati raggruppati in un apposito canale Discord in cui vengono mandati tutti gli acquisti effettuati dagli utenti e contati tramite apposito strumento di ricerca offerto dalla piattaforma stessa. I 328 risultati garantiscono per i 35 utenti attuali una media di circa 10 paia di sneakers di Jordan 1 Mid a testa nei 6 mesi di sviluppo.

Infine, una volta usciti dalla fase si beta, si prevede sia di aumentare la base di utenti sia di sviluppare nuovi algoritmi in modo da eludere le restanti soluzioni di protezioni da bot esistenti. Nel prossimo futuro si conta inoltre di sviluppare una interfaccia grafica (GUI) accattivante basata su JavaFX e di integrarla con gli algoritmi Python tramite un apposito Framework.

Bibliografia

1. Comunicazione della Commissione Europea COM(97) 157 del 15 aprile 1997
2. Philip Evans, Thomas S. Wurster, Strategy and new economics of Information Strategy and new economics of Information, 1997
3. Bauer, Raymond, Consumer Behavior as risk taking, 1960
4. David A. Aaker, Erich Joachimsthaler, Brand Leadership, 2009
5. Luiz Moutinho, Pedro Dionísio, Carmo LealCarmo Leal, Surf tribal behaviour: A sports marketing application, 2007
6. Adrian FurnhamDimitrios Tsivrikos, A Psychologist in the Marketplace, 2016
7. Marketing to Millennials: How Luxury Brands Build Hype, <https://luxedigital.digital-luxury-trends/marketing-hype-millennials>
8. Neal Heard, Sneakers, 2008
9. Michael Lishnevsky, The rise of the sneakerhead culture, <https://www.ix.co/en-US/blog/the-rise-of-the-sneakerhead-culture>
10. Thorstein Veblen, Critical Assessments Volume 1
11. <https://www.complex.com/style/2013/01/hype-a-mathematical-equation>
12. Marketing to Millennials: How Luxury Brands Build Hype, <https://luxedigital.digital-luxury-trends/marketing-hype-millennials>
13. L'evoluzione del concetto di lusso e la gestione strategica della marca. Un'analisi qualitativa delle percezioni sul concetto, sulla marca e su un prodotto di lusso. Gaetano Aiello Professore Straordinario di Marketing Facoltà di Economia, Università degli Studi di Firenze, Raffaele Donvito*, Dottore di Ricerca, Facoltà di Economia, Università degli Studi di Firenze, CONGRESSO INTERNAZIONALE "LE TENDENZE DEL MARKETING"
14. <https://draftinmag.net/2019/03/21/supreme-e-lhype-di-james-jebbia/>
15. Streetwear gets a luxe makeover as heritage brands aim to influence millennials, <https://www.scmp.com/magazines/style/fashion-beauty/article/2123433/streetwear-gets-luxemakeover-heritage-brands-aim>
16. Sussman, The exception is the rule: Underestimating and overspending on exceptional expenses. Journal of Consumer Research, 2012
17. Bernard Cova,Alex Giordano,Mirko Pallera, Marketing non-convenzionale. Viral, guerrilla, tribal, societing e i 10 principi fondamentali del marketing postmoderno , 2011

18. Brand e distribuzione: cosa ci insegna il caso di Supreme, <https://www.antartika.it/brand-e-distribuzione-cosa-ci-insegna-il-caso-di-supreme/>, 2018
19. A History of Supreme: From Independent Skate Shop to Global Powerhouse, https://www.klekt.com/blog/a-history-of-supreme-from-independent-skate-shop-to-global-powerhouse_i1045, 2019
20. Inside Supreme: “Anatomy of a global streetwear cult – Part 2”, <https://hypebeast.com/2012/1/the-business-of-fashion-inside-supreme-anatomy-of-a-global-streetwear-cult-part-2>, 2012
21. Supreme’s Midas Touch – DJ Ross Wilson Discusses the Genius Behind the Label, <https://www.sothebys.com/en/articles/supremes-midas-touch-dj-ross-wilson-discusses-the-genius-behind-the-label>, 2019
22. Ecco perchè secondo noi il bot-game di Supreme non ha (quasi) più senso, <https://iobimag.com/talks/perche-bot-game-supreme-non-ha-senso/>, 2019
23. [https://it.wikipedia.org/wiki/Bot_\(informatica\)](https://it.wikipedia.org/wiki/Bot_(informatica))
24. How Is an Internet Bot Constructed?, <https://www.cloudflare.com/learning/cloud/how-is-an-internet-bot-constructed/>
25. Diletta Parlangeli, I bot generano oltre la metà di tutto il traffico di internet, <https://www.wired.it/internet/web/2017/11/02/bot-generano-meta-traffico-di-internet/>, 2017
26. Dunham, Ken; Melnick, Jim, Malicious Bots: An Inside Look into the Cyber-Criminal Underground of the Internet, 2018
27. Zeifman, Igal, Bot Traffic Report 2016, 2017
28. https://en.wikipedia.org/wiki/Sneaker_collecting
29. <https://www.theshitbot.com/sneaker-cookgroups/>
30. <https://www.theshitbot.com/sneaker-cookgroups/>
31. Scalper bot protection: How to protect your e-commerce site against scalping, <https://datadome.co/bot-management-protection/bot-protection-for-scalping-the-automated-owasp-threat-targeting-e-commerce/>
32. Jeremy D'Hoinne, Adam Hils, Ayal Tirosh, Claudio Neva, Magic Quadrant di Gartner per i Web Application Firewall, 2018
33. <https://pypi.org/project/cloudscraper/>
34. Matthew Prince, Sergi Isasi, Moving from reCAPTCHA to hCaptcha, <https://blog.cloudflare.com/moving-from-recaptcha-to-hcaptcha/>, 2020
35. <https://www.akamai.com/it/it/campaign/assets/reports/2020-forrester-new-wave-bot-management.jsp>

36. Scalper bot protection: How to protect your e-commerce site against scalping,
<https://datadome.co/bot-management-protection/bot-protection-for-scalping-the-automated-owasp-threat-targeting-e-commerce/>
37. Pietro Riccucci, A look into Off White Jordan 4 Drop,
<https://medium.com/@notpietro1234/a-look-into-off-white-jordan-4-drop-1205cc03fcbd>
38. <https://docs.python.org/3/library/logging.html>
39. <https://support.discord.com/hc/it/articles/228383668-Come-usare-i-Webhooks>
40. <https://support.discord.com/hc/it/articles/228383668-Come-usare-i-Webhooks>
41. <https://portswigger.net/web-security/csrf/tokens>
42. <https://pypi.org/project/cloudscraper/>
43. <https://requests.readthedocs.io/en/master/>
44. <https://www.html.it/pag/15614/dizionari/>
45. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
46. <https://pypi.org/project/cloudscraper/>
47. <https://selenium-python.readthedocs.io/>
48. <https://docs.python.org/3/library/threading.html>
49. <https://www.html.it/pag/69486/multithreading/>
50. <https://pypi.org/project/discord-webhook/>

Riferimenti generali

Giovanni Fracasso, I software eCommerce più usati e relative considerazioni,
<https://www.ictsviluppo.it/e-commerce/software-e-commerce-pi%C3%B9-usati-considerazioni>, 2019

Michela Maggi, E-commerce: che cos'è e perché è utile alle aziende?,
<http://www.diritto24.ilsole24ore.com/avvocatoAffari/mercatiImpresa/2012/11/e-commerce-che-cos-e-e-perche-e-utile-alle-aziende.php>, 2019

Che cos'è l'e-commerce? Definizione di e-commerce per il 2020, <https://ecommerce-platforms.com/it/glossary/ecommerce>

Storia dell'e-commerce: origini ed evoluzione del commercio elettronico,
<https://www.gustorotondo.it/storia-dell-e-commerce/#!exploit-e-commerce>, 2018

Daniella Niyonkuru, ChatOps at Shopify: Inviting Bots in Our Day-to-Day Operations,
<https://www.usenix.org/conference/lisa17/conference-program/presentation/niyonkuru>, 2017

Gentian Hajdaraj, E-commerce significato e l'importanza delle vendite per la tua azienda,
<https://leadcommerce.it/e-commerce-significato/>, 2018

Colombo R., Control and Automation Systems, 2011

La nozione di commercio elettronico, <http://www.dirittodellinformatica.it/e-commerce/ecommerce-focus/nozione-commercio-elettronico.html>, 2017

<https://trends.builtwith.com/shop/Shopify>

Thames & Hudson, This is not fashion: streetwear past, present and future, 2018

J. H., PINE, B. J, Autenticità. Ciò che i consumatori vogliono davvero, 2009

Why isn't streetwear just called fashion?, <https://qz.com/quartz/1160897/why-isnt-streetwear-just-called-fashion/>, 2018

Supreme, il segreto del successo mondiale, <http://www.mywhere.it/supreme-segreto-del-successo-mondiale/>, 2018

Why Kim Jones brings punk-influenced streetwear to Louis Vuitton,
<https://www.scmp.com/magazines/>, 2018

[style/people-events/article/2087647/why-kim-jones-brings-punk-influenced-streetwear-louis](https://www.scmp.com/magazines/style/people-events/article/2087647/why-kim-jones-brings-punk-influenced-streetwear-louis-vuitton)>

Why Are So Many People Obsessed with Supreme,
https://www.vice.com/en_us/article/5gq393/supreme-and-the-psychology-of-brand-devotion, 2016

The hype machine. Streetwear and the business of scarcity,
<http://www.bbc.com/capital/story/20180205-the-hypemachine-streetwear-and-the-business-of-scarcity>, 2018

Perché il 2017 è l'anno in cui lo streetwear è diventato mainstream?, <https://i-d.vice.com/it/article/ywn3qv/streetwear-da-sottocultura-a-fenomeno-mainstream>, 2017

Inside Supreme: Anatomy of a Global Streetwear Cult — Part I,
<https://www.businessoffashion.com/articles/intelligence/inside-supreme-anatomy-of-a-global-streetwear-cult-part-i>, 2012

Inside Supreme: Anatomy of a Global Streetwear Cult — Part II,

<https://www.businessoffashion.com/articles/intelligence/inside-supreme-anatomy-of-a-globalstreetwear-cult-part-ii>, 2012

Building Brands Without Mass Media, <https://hbr.org/1997/01/building-brandswithout-mass-media>

Is Luxury's Love Affair With "Drops" More Than Marketing?, <https://www.businessoffashion.com/articles/professional/is-luxurys-love-affair-with-supreme-style-drops-more-than-marketingmoncler-burberry-off-white-celine>, 2017

Cult streetwear label Supreme: Why fans go nuts over these clothes, <https://www.news.com.au/lifestyle/fashion/designers/cult-streetwear-label-supremewhy-%20fans-go-nuts-over-these-ugly-clothes/newsstory/%209f961920a4843b2db1c2d0ff30e635d5>, 2017

What Is Bot Traffic? | How to Stop Bot Traffic, <https://www.cloudflare.com/learning/bots/what-is-bot-traffic/>

How Is an Internet Bot Constructed?, <https://www.cloudflare.com/learning/cloud/how-is-an-internet-bot-constructed/>

Matthew Prince, Sergi Isasi, Moving from reCAPTCHA to hCaptcha, <https://blog.cloudflare.com/moving-from-recaptcha-to-hcaptcha/>, 2020

Variti's Bot analytics, <https://www.cloudflare.com/apps/variti-bot-analytics>

Michael Kan, Cloudflare Dumps Google's ReCAPTCHA Over Privacy Concerns, Costs, <https://www.pcmag.com/news/cloudflare-dumps-googles-recaptcha-over-privacy-concerns-costs>, 2020

Mitchell Anicas, How To Mitigate DDoS Attacks Against Your Website with CloudFlare, <https://www.digitalocean.com/community/tutorials/how-to-mitigate-ddos-attacks-against-your-website-with-cloudflare>, 2015

<https://www.akamai.com/it/it/products/security/>

<https://www.akamai.com/it/it/multimedia/documents/product-brief/bot-manager-product-brief.pdf>

Scalper bot protection: How to protect your e-commerce site against scalping, <https://datadome.co/bot-management-protection/bot-protection-for-scalping-the-automated-owasp-threat-targeting-e-commerce/>

<https://datadome.co/bot-management-protection/>

I'm (really) not a robot: Captcha farms and the challenges of Captcha bot detection,

<https://datadome.co/bot-detection/how-to-detect-captcha-farms-and-block-captcha-bots/>

<https://www.perimeterx.com/solutions-by-industry/e-commerce/>

Six Major Threats to E-commerce and How to Stop Them,

<https://www.perimeterx.com/resources/whitepapers/six-major-threats-to-ecommerce/>

<https://variti.com/uk-en/services/api-protection/>