UNIVERSITÀ POLITECNICA DELLE MARCHE
CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

Analisi dei dati per la previsione delle tendenze nel settore fashion

Data analysis for forecasting trends in the fashion industry.

Relatore:
Emanuele Frontoni

Tesi di Laurea di:
Riccardo Medici

Correlatore:
Marina Paolanti

A.A. 2021/2022

Sommario

L’avvento di Internet, oltre a tutte le sue funzioni che utilizziamo giornalmente, ha portato alla creazione della più grande fonte di dati della storia. Chiunque utilizzasse dati storici e tecniche statistiche per generare delle previsioni ora ha accesso ad un pool di dati immenso. Questa nuova branca dello studio dei dati si chiama Data Science e per accedere ai dati presenti su internet esistono tecniche e metodologie specifiche. Una di queste tecniche viene trattata in questa tesi di laurea. La tecnica portata come esempio consiste in un “web Scraper”, ovvero un programma utilizzato per raccogliere informazioni dai siti internet. Nello specifico, questo programma emula il comportamento che avrebbe un utente umano e visualizza gli elementi presenti su Farfetch, un E-Tailer online, dove sono presenti borse, scarpe e altri oggetti del mondo della moda. Dagli oggetti visualizzati vengono poi estrapolate delle informazioni come, ad esempio, il prezzo o le dimensioni. Questi dati vengono poi memorizzati in un file Json così che, tramite delle tecniche di manipolazione di dati, sia possibile generare informazioni utili riguardo i brand e le tipologie di oggetti presenti sul sito.

Indice

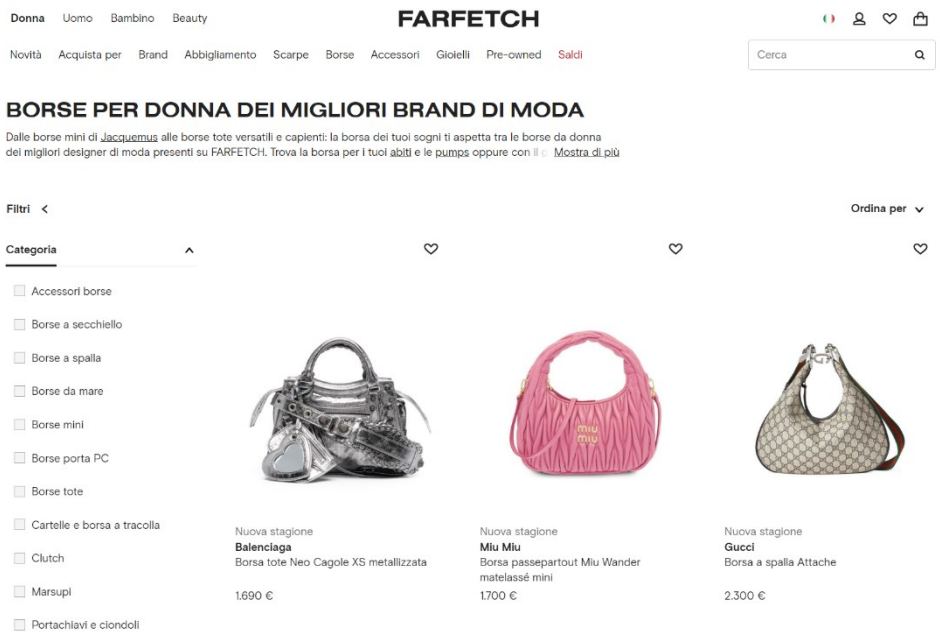
1 Introduzione

1.1 Contesto

Essere in grado di prevedere il futuro è sempre stata un sogno di noi esseri umani, nei tempi antichi ci si affidava ai rituali, mentre nel tempo recente, ci si è avvicinati sempre di più alla matematica e alla statistica. La limitazione di queste metodologie, è che per ottenere delle previsioni precise è essenziale avere una grande quantità di dati storici su cui basarsi. Tuttavia, considerando la scarsità di dati disponibili e facilmente accessibili, ottenere ed estrapolare informazioni da questi dati non era un compito facile. Al giorno d'oggi invece, internet rappresenta un enorme banca dati facilmente accessibile, generando dati che possono essere agevolmente elaborati grazie a numerosi software di analisi e alla nascita della così detta *data science.* Queste tecniche di estrapolazioni dati vengono usate in molti settori diversi, dalla scienza alla finanza. In questa tesi di laurea ci si concentrerà invece sulle analisi di mercato. Le aziende che operano in questo settore, sempre più competitivo e globalizzato, necessitano infatti della data analisi per ottenere previsioni precise per ottenere un vantaggio rispetto alla concorrenza.

1.2 Obbiettivi

L'obbiettivo della tesi è lo sviluppo di un software capace di estrapolare informazioni dal sito Farfetch[1] per poi utilizzare i dati ricavati per effettuare delle analisi sui trend e l'attuale distribuzione del mercato delle borse di lusso. Per ottenere questo risultato è stato sviluppato un web scraper, un software che imitando il comportamento umano, naviga il sito web di Farfetch[1] visualizzando e raccogliendo di volta in volta informazioni sui vari articoli in vendita.



Porzione del sito Farfectch.com

1. https://www.farfetch.com/it/shopping/women/items.aspx

1.3 Struttura della tesi

La tesi può essere divisa in due sezioni distinte, nella prima parte gli argomenti trattati riguardano il contesto, le applicazioni della data analisi e le varie tecniche disponibili con i loro vantaggi e svantaggi. In seguito, nel capitolo 3 e 4, si entra più nel dettaglio nello sviluppo del codice utilizzato nel progetto: vengono prima esposti i tools principali utilizzati per poi esporre più nello specifico il funzionamento del codice. Infine, al capitolo 5 i dati analizzati vengono utilizzati per generare dei grafici, le informazioni ricavabili da essi verranno poi commentate.

2 Stato dell'arte

2.1 Importanza della data analisi

Gli individui responsabili di una qualsiasi attività lucrativa, quando in dubbio su quali opzioni potessero essere più convenienti per la loro attività, sono sempre stati portati, anche inconsciamente, a fare delle ricerche per prevedere l'andamento futuro del mercato. Sebbene questa pratica sia utilizzata da migliaia di anni, lo studio effettivo delle modalità di ricerca è piuttosto recente [1].

A prescindere dalla grandezza della propria attività, nel mercato globalizzato ed estremamente competitivo in cui viviamo, è diventato apparente come le informazioni e i dati siano l'arma più importante per scegliere [2] e sfruttare le migliori opportunità.

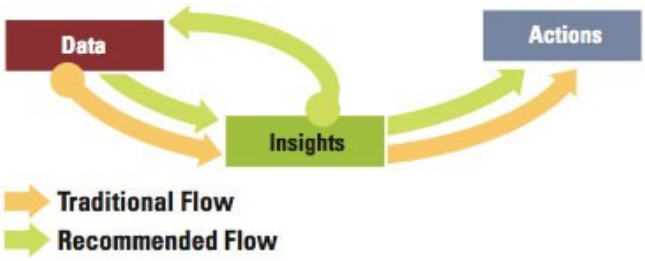


Figure 1 Come l'analisi dati cambia il processo di decision making

Come scritto nello studio pubblicato dal MIT [2] le aziende più performanti, infatti, utilizzano studi la data analisi cinque volte tanto rispetto ai loro concorrenti meno efficaci. Per questo motivo ovviamente lo studio dei dati è diventata la priorità per moltissime aziende, tuttavia, adottare queste soluzioni comporta anche degli ostacoli. Sempre secondo lo studio MIT [2], la maggior parte dell'aziende interessate non ha percepito come maggior ostacolo la complessità dell'analisi dati o la qualità dei dati stessi. Al contrario, le barriere all'adozione di queste metodologie sono soprattutto di carattere manageriale e culturale. Infatti, il più grande

problema risiede nel fatto che la classe dirigenziale di una azienda non riesce a comprendere le potenzialità dell'analisi dati e continua a basarsi sull'intuito e sulla propria esperienza.

Un modo per rendere più fruibile la data analisi sono tutti quei tool, i quali stanno diventando sempre più efficaci per esprimere graficamente le conclusioni ricavate dai dati.

Queste includono la visualizzazione dei dati, analisi ricavate dai social media e simulazioni. In questo modo si trasformano numeri in informazioni per essere poi utilizzate nel processo decisionale.

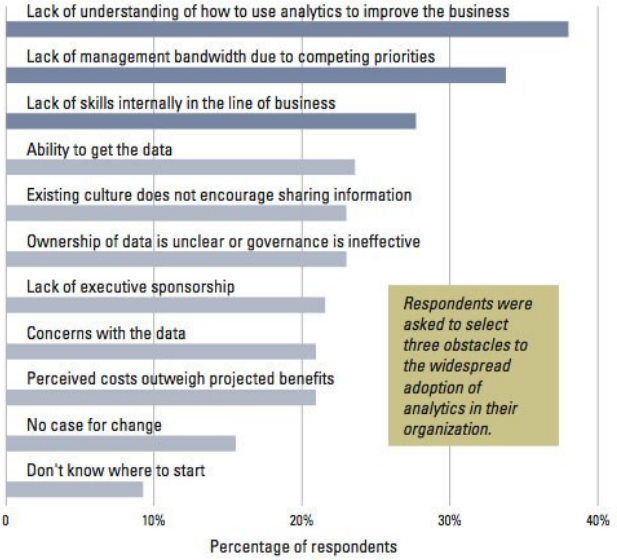


Figure 2 Barriere per l'adozione di tecniche di data analisi

2.2 Business Intelligence

Si definisce Business Intelligence l'utilizzo di tecniche e pratiche che analizzano i dati disponibili per aiutare le imprese a prendere decisioni basandosi sulle previsioni ricavate dall'analisi dati.
Poiché questi metodi dipendono fortemente dai dati, la loro estrazione e manipolazione è estremamente importante. I metodi analitici usati in questi sistemi per ottenere delle previsioni si basano su tecniche statistiche e sull'estrazione di dati.

Figure 3 Scenario di business analytics

I vecchi sistemi di business intelligence si basavano per lo più sull'analisi dei dati storici, dai quali non era spesso possibile estrarre previsioni accurate. Dopo i primi anni 2000 invece, l'utilizzo di internet mise a disposizione delle aziende una enorme quantità di dati da cui attingere. Da allora, le aziende iniziarono ad utilizzare tecniche e software sempre più sofisticati per sfruttarli. Come riportato detto in precedenza, riproponendo l'articolo del MIT, le metodologie che utilizzano i Big Data, termine usato per descrivere enormi volumi di dati, hanno dimostrato di poter generare previsioni molto più accurate e utili.

Le aziende che per prime hanno utilizzato queste tecniche sono ovviamente, per le loro capacità e competenze, le “big tech” americane. Due esempi sono Google e Amazon i due colossi americani, creando sistemi previsionali efficienti, hanno infatti ottenuto un grande vantaggio rispetto ai loro competitors. Il motore di ricerca Google, infatti, è stato il primo ad implementare efficacemente algoritmi che prevedevano le ricerche dell’utente, proponendo quindi i migliori risultati. Ugualmente Amazon, quando ha introdotto un algoritmo capace di prevedere le necessità del cliente e di proporre prodotti potenzialmente interessanti, ha incrementato il proprio utile del 30%. [3]

Le stesse tecniche utilizzate da Google e Amazon ormai vengono utilizzate in molti settori diversi, dove le aziende più performanti riescono ad ottenere dei vantaggi rispetto ai loro concorrenti grazie all’ utilizzo della data analisi e dell’analisi previsionale. Per capire meglio come la data analisi è utile alle aziende, è opportuno fare qualche esempio di ambiti di applicazione.

- **Marketing:** L’acquisizione di dati è un elemento essenziale per poter creare campagne pubblicitarie efficaci a seconda del target di clienti a cui si è interessati. Al giorno d’oggi è possibile raccogliere dati di ogni tipologia su un individuo, sia a livello demografico che di preferenze personali. L’utilizzo dei cosiddetti “cookies” nelle pagine web è uno dei metodi utilizzati per raccogliere informazioni che verranno poi utilizzate per produrre pubblicità mirate.
- **Manifattura:** Una analisi dei dati relativi alle prestazioni e ai processi delle varie stazioni di un impianto produttivo, permette di elaborare sistemi che siano il più efficaci possibili. Un esempio concreto sono i sistemi produttivi della cosiddetta Industria4.0, dove i macchinari generano dati sul proprio funzionamento che poi vengono elaborati per prevenire guasti e blocchi nella produzione.
- **Finanza:** Uno dei settori dove la data analisi ha avuto gli impatti maggiori in quanto l’elaborazione di dati è il metodo più efficace per prevenire i mercati finanziari.

Come è stato scritto nel precedente paragrafo, lo studio dei dati è essenziale per essere competitivi sul mercato. Questi dati possono essere di varia natura: possono essere di natura sociale, riguardando quindi il personale, altrimenti possono essere relativi ai tempi di utilizzo dei macchinari, oppure possono essere relativi ai prodotti in vendita nel mercato. Come riportato precedentemente Internet è una delle fonti più utilizzata dalle aziende per reperire i dati, e in questa tesi di laurea verrà presentato un esempio di estrapolazione dati da una pagina web, tramite l’uso del “web scraping”.

2.3 Web Scraping

Il web scraping, anche conosciuto come web extraction [4], è una tecnica utilizzata nella data analisi per estrarre dati da un sito web per poi salvarli in un file o un database per poi estrapolarli ed analizzarli.

Questo processo avviene manualmente o tramite l'utilizzo di software automatizzato, chiamato per l'appunto *scraper*. Il processo di raccogliere i dati da internet può essere diviso nella successione di due azioni: l'acquisizione della pagina dove si trovano i dati e la loro estrapolazione. Più nello specifico uno scraper inizia dal fare una richiesta al sito di accedere alle informazioni presenti. Questa richiesta viene solitamente mandata tramite un URL contenente una stringa di ricerca. Una volta che la richiesta viene ricevuta e processata dalla corrispettiva pagina web, i dati vengono estrapolati e copiati in formato XML o JSON[5].

Figure 4 Raffigurazione fasi web scraping

Come pubblicato nell'articolo presentato alla conferenza ICECA 2019[6] ci sono differenti approcci per costruire uno scraper ognuno con i propri vantaggi e svantaggi:

- **Metodo Mimico**

  Questa categoria di scraper funziona grazie a delle regole prefissate impostate dall'utente. La locazione dei dati, ovvero dove essi sono salvati è preconfigurata nel codice, il programma infatti utilizza l'indirizzo Xpath o il nodo html dell'elemento per localizzarlo (paragrafo 3.3). Ciò comporta che questo metodo non sia molto adatto per siti con pagine che hanno un layout non costante, in quanto l'indirizzo Xpath o nodo html muterebbero. Inoltre, anche qualora il sito scelto cambiasse il suo aspetto, anche il solo lato estetico, sarebbe necessaria una riprogrammazione del codice per lo stesso motivo.

- **Weight Measurement Approach**

  Questo approccio consiste in un algoritmo generico che analizza il DOM tree di un documento html e a seconda del peso delle parole in ogni ramo del DOM tree, l'algoritmo sceglie quali dati estrapolare. Questo approccio può essere utilizzato solo in determinati contesti dove non è importante il contenuto del singolo dato bensì la quantità di volte che viene estrapolato.

- **Metodo differenziale**

  Questo metodo consiste nel fare una comparazione fra pagine dello stesso sito ed eliminare tutti gli elementi in comune. In questo modo elementi non contenenti dati come le finestre di navigazione o le colonne laterali vengono

eliminate, lasciando solo i dati che si differenziano che generalmente sono quelli contenenti le informazioni.

- **Machine Learning**

  Il concetto base di questo metodo è di allenare un algoritmo su un campione esteso di pagine web manualmente analizzate. Il machine learning è basato su indicatori spaziali che aiutano l'algoritmo a dedurre dove si trova il testo da estrapolare.

2.4 Comparazione performance dei metodi Regex, Xpath, HTML dom.

Nel contesto di questo progetto è stato preferito un metodo mimico (*mimicry method*) in quanto rispecchiava meglio le esigenze del compito che doveva svolgere. Dovendo estrarre dati specifici da un sito, il metodo differenziale e il *Weight Measurament Approach* non erano indicati poiché avrebbero prodotto risultati poco precisi. Inoltre, l'azione di estrapolare i dati è avvenuta solo nell'arco di qualche giorno, evitando il problema di dover aggiornare il codice qualora il sito avesse subito dei cambiamenti.

Anche all'interno del *mimcry method* ci sono diversi metodi per selezionare ed estrapolare il singolo dato o elemento. Un paper di ricerca pubblicato su "Atlantis Press" ha messo a confronto le prestazioni di tre metodi: Xpath, HTML DOM, Regex.

- **Xpath**
  Xpath è un indirizzo univoco proprio di ogni elemento in una pagina HTML. Tramite gli Xpath è possibile navigare ed accedere ad ogni elemento nella pagina.
- **HTLM DOM**
  L'Hyper Text Markup Language Document Object Model è uno standard per accedere, modificare e cancellare elementi HMTL. HTML DOM usa linguaggi di programmazione come JavaScript e Python per accedere agli elementi html.

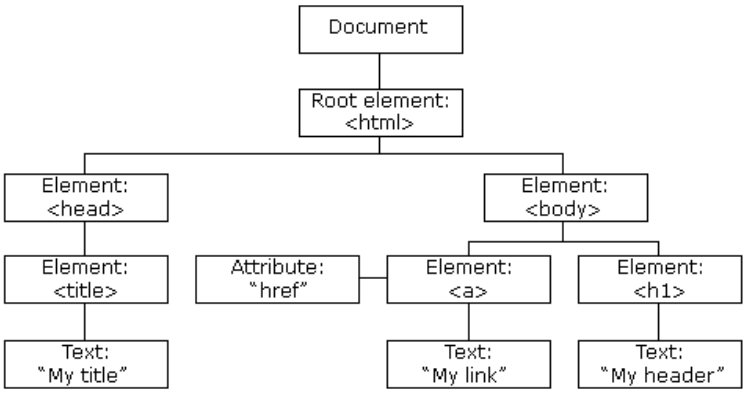


Figure 5 Raffigurazione HTML DOM

- **Regular expression (Regex)**
  Le espressioni regolari sono un metodo per descrivere e identificare un determinato gruppo di parole. In questo modo è possibile filtrare i risultati e trovare solo quelli relativi a quel "set" di parole. "*Supponiamo che questi indirizzi IP siano compresi nell'intervallo 198.51.100.1 - 198.51.100.25. Invece di inserire 25 indirizzi IP diversi, puoi creare un'espressione regolare come 198\.51\.100\.\d* che corrisponde all'intero intervallo degli indirizzi.*
  In alternativa, se volessi creare un filtro vista che includesse solo dati della campagna provenienti da due città diverse, potresti creare un'espressione regolare come San Francisco|New York (San Francisco o New York)." (Guida di Google Analytics)

Per misurare le performance dei tre metodi, i ricercatori hanno realizzato un esperimento così composto. Inizialmente, viene caricata pagina web utilizzando un browser e vengono scelti alcuni elementi che dovranno essere identificati con i tre metodi. Il codice usato consiste nella creazione di un Web Scraper, scritto in Java, che accede agli elementi specificati e gli stampa a schermo. Si valutano quindi le performance dei tre metodi, misurando il tempo impiegato e la memoria utilizzata.

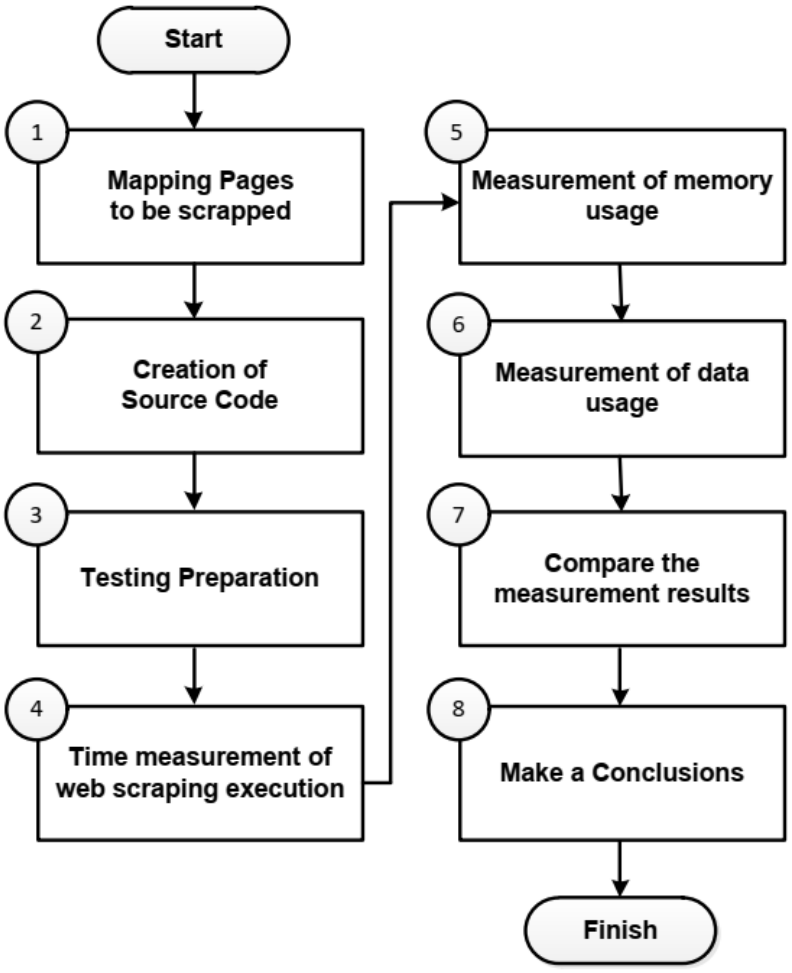


Figure 6 Rappresentazione degli step del test

Dai risultati dei testi emerge che il metodo Regex ha ottenuto un tempo medio di risposta di **399.75ms**, the DOM HTML method ha una media di **298.55ms** e il metodo Xpath ha una media di **435.15ms**.

Experiment	Time (ms)		
	REGEX	HTML DOM	XPATH
1	375	297	406
2	375	250	407
3	390	360	485
4	391	281	859
5	391	454	422
6	382	265	390
7	446	266	406
8	322	281	422
9	325	265	406
10	377	500	438
11	294	250	391
12	286	265	375
13	422	266	640
14	390	266	390
15	516	250	375
16	406	266	375
17	406	250	375
18	594	282	375
19	391	407	391
20	516	250	375
Avg	**399,75**	**298,55**	**435,15**

Per quanto riguarda le performance relative alla memoria utilizzata all'esecuzione dello Web Scraper per ogni metodo si ottengono risultati analoghi infatti, infatti il minor utilizzo di memoria si ottiene con il metodo Regex: **564KB**, seguito dal metodo Xpath con **574KB** e infine da HTML DOM **4.8MB**.

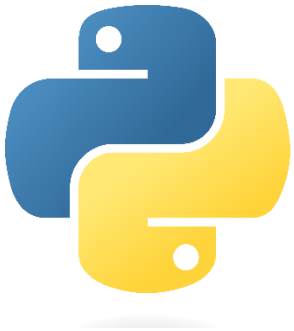
Experiment	Memory Usage (bytes)		
	REGEX	HTML DOM	XPATH
1	513.264	4.699.048	713.320
2	664.416	4.739.912	505.904
3	625.552	4.614.448	625.584
4	461.368	5.084.552	486.008
5	572.576	4.707.232	923.528
6	592.123	4.618.816	477.800
7	722.345	4.743.344	461.400
8	772.364	4.703.696	584.720
9	547.326	4.743.400	694.072
10	682.483	4.730.112	461.400
11	469.576	5.098.016	656.416
12	485.976	4.684.760	584.232
13	505.176	5.090.696	469.608
14	489.472	5.091.936	584.752
15	485.976	5.087.408	462.576
16	461.416	4.652.704	461.400
17	461.368	4.679.448	625.672
18	461.368	5.084.200	625.584
19	664.496	4.836.776	625.552
20	657.008	4.652.144	461.400
Avg	**564.782,5**	**4.817.132**	**574.546,4**

Nel contesto di questo progetto è stato scelto il metodo Xpath poiché rappresenta un buon compromesso tra tempi di risposta, occupazione di memoria e complessità di implementazione.

3 Materiali e metodi

In questo capitolo vengono descritte le tecnologie, gli strumenti e le librerie utilizzate durante lo sviluppo del progetto. Per ognuno di essi viene descritto il motivo della scelta e dove viene utilizzato all'interno del progetto.

3.1 Phython



Python è un linguaggio multiuso, ovvero che può essere usato per una grande varietà di applicazioni, questa sua caratteristica lo rende uno dei linguaggi più usati dai programmatori. Secondo un sondaggio di Stack overflow [7], infatti, il 41,7% degli intervistati utilizza abitualmente Python.

Tra i vantaggi che offre Python rispetto agli altri linguaggi di programmazione risaltano in primis la facilità d'uso, la sua sintattica semplice e di alto livello lo rendono molto indicato per utenti poco esperti che si stanno approcciando alla programmazione.

Inoltre, motive per il quale viene utilizzato in questo progetto, Python dispone di una grande disponibilità di librerie che incrementano ancor di più i suoi contesti di utilizzo, come ad esempio:

- Flask: Flask è una libreria che permette lo sviluppo di applicazioni ad-hoc per siti web.
- TensorFlow: TensorFlow è una libreria sviluppata da Google per il calcolo computazionale.
- Selenium: Selenium è la libreria utilizzata in questo progetto. Il suo principale utilizzo è il *web scraping,* le funzioni di cui dispone infatti, sono in grado di interagire con le pagine web ed estrarre dati. Questa libreria verrà descritta nel dettaglio in seguito.

Attualmente ci sono due versioni attive di Python:

- Python 2, rilasciata nel 2000 e aggiornata alla sua ultima versione, la 2.7 nel 2010
- Python 3, rilasciata nel 2008 e continuamente aggiornata con un intervallo di 17 mesi.

La caratteristica di Pyhton di essere un linguaggio ad alto livello e con una sintattica più vicina all'inglese rispetto ad altri linguaggi di programmazione fanno sì che non sia molto efficace per quanto riguarda l'utilizzo della memoria. Per questo motivo Python non è molto popolare quando la scalabilità, ovvero la caratteristica di usare lo stesso codice in macchine diverse, anche con potenza di calcolo ridotte, è una caratteristica importante.

3.2 Librerie di Python utilizzate

3.2.1 Selenium

Selenium è un progetto figlio del lavoro di decine di volontari che collaborando, hanno creato questa libreria gratuita open source, che permette agli utenti di interfacciarsi con un browser.

La caratteristica principale di Selenium è l'interfaccia WebDriver, ovvero la possibilità di utilizzare un browser emulando il comportamento umano, rendendolo intuitivo da utilizzare sia per testare le pagine web sia per creare tools di web scraping. Questa sua funzionalità è stato il motivo principale per il quale è stato utilizzato in questo lavoro. Tra i vantaggi che offre, oltre alla sua natura gratuita e open source, è la possibilità di utilizzare praticamente tutti i browser più popolari tra cui: Chrome, Firefox, Edge, Opera e Safari.

3.2.2 Pandas

Pandas è una libreria Python open source utilizzata per la manipolazione dei dati e la loro analisi. Citando il loro sito pandas (che sta per **pan**el **da**ta) "*mira ad essere l'elemento principale per eseguire analisi pratiche, dei dati del mondo reale in Python. Inoltre, ha l'obiettivo più ampio di diventare lo strumento di analisi/manipolazione dei dati open source più potente e flessibile disponibile in qualsiasi lingua*."

Riassumendo vuole quindi essere il principale tool per la visualizzazione e manipolazione dati in Python e non solo. Pandas offre infatti la possibilità di leggere dati da file di diversi formati per poi, tramite le sue varie funzionalità, manipolarli nel modo più modo più chiaro ed efficace possibile. Pandas funziona utilizzando delle *data structure*, ovvero delle tabelle nelle cui celle i dati letti dai file vengono memorizzati per poi essere manipolati.

Inoltre, questa libreria open source permette di creare diverse tipologie di grafici e tabelle in modo molto più rapido rispetto a pratiche più tradizionali come Excel, per questo motivo viene molto utilizzato per visualizzare graficamente le conclusioni che possono essere tratte da una analisi di dati. Pandas è estremamente versatile, infatti come detto in precedenza è in grado di leggere di molti formati diversi tra cui: CSV, JSON, HTML, EXCEL, SQL, PARQUET e LaTeX. Nel contesto di questa tesi è stato utilizzato Pandas per leggere i dati presenti nel file Json e creare i grafici presenti al capitolo 5.

3.2.3 Time e datetime

Le librerie time e datetime, aggiunte al codice tramite la funzione import, sono due librerie standard presenti in ogni programma a prescindere dalla sua applicazione. La libreria time permette infatti di utilizzare funzioni che prendono come argomento un intervallo di tempo. Sono estremamente utili per ogni applicazione dove è necessario far trascorrere un determinato intervallo di tempo per chiamare una funzione. La libreria datetime invece, permette accedere alle informazioni di data e orario in diversi formati. Nello specifico di questo progetto viene utilizzata per sapere quando le informazioni di un singolo componente sono state estratte.

3.2.4 Json

Json è una abbreviazione per JavaScript Object Notation, ed è un formato per la scrittura e trasporto dei dati. Il formato Json viene utilizzato per la sua leggerezza e facilità di utilizzo sia per la macchina che per l'utente, per questo motivo è particolarmente efficace se i dati vengono analizzati da un server o una pagina web come nel nostro caso.

Il formato Json è sintatticamente uguale al codice utilizzato per creare oggetti in java script. Proprio per questa similarità sia Java Script che Pyhton posson facilmente convertire i dati, rendendo scrivere e leggere i file Json particolarmente efficace.

3.2.5 Webdriver

Webdriver è un programma open source automatizzato per testare applicazioni web su diversi browser. Tra le funzioni offre la possibilità di navigare le pagine web, user inputs, esecuzioni di codice Java Script e altro ancora. Nello specifico di questo progetto è stato usato ChromeDriver, una delle varianti di WebDriver, che si interfaccia con il browser Google Chrome sia su android che su desktop.

Il programma viene scaricato nel desktop e tramite una specifica funziona va chiamato per essere utilizzato all'interno del programma come viene descritto in seguito.

3.3 Definizioni utili

In queta sezione verranno riportate alcune terminologie utili per comprendere più facilmente il contenuto all'interno di questo report.

- **XPATH**: Xpath sta per XML Path Language. Usa una sintassi che ricorda per l'appunto un path, ovvero una traccia, che "naviga" i nodi in un path xml e identifica univocamente un elemento in una pagina web.
- **URL**: Uniform Resource Locator, rappresenta l'indirizzo di una risorsa su Internet e i protocolli utilizzati per accedervi
- **DOM Tree**: il DOM, Document Object Model, è una forma di rappresentazione dei documenti. Nel contesto dei documenti html il DOM è un modo per aggiornare dinamicamente il contenuto, la struttura e lo stile dei documenti.

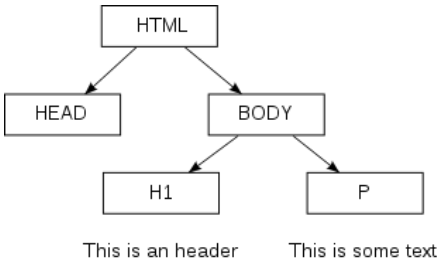


Figure 7 Rappresentazione di una struttura DOM

- **Array:** Si definisce array un insieme di variabili, tutte dello stesso tipo, identificate da un nome unico.
- **Dizionario:** I dizionari sono un insieme di dati (stringhe, liste, interi), che si differenziano per la presenza di chiavi e valori. In un dizionario le chiavi sono uniche e ad ogni chiave è associata uno o più valori.

```
entry = {
    "brand": brand.text,
    "bag type": bag_type.text,
    "price": priceINT,
    "BrandID": IDbrandtext,
    "bag_url":bag_url,
    "description": descrizione,
    "material": materiali,
    "dimensions": dimensioni,
    "bag_image": bag_image,
    "date and time": time_date,
}
```

Figure 8 Esempio di dizionario usato nel progetto

- **.driver:** variabile che verrà utilizzata durante il codice per utilizzare le funzioni presenti nel tool chromedriver, essa infatti è associata alla funzione webdriver.Chrome, funzione che per l'appunto prende come argomento la locazione della memoria dove è salvato il tool (PATH).

3.4 Metodi principali

- driver.get() → La funzione .get() prende come argomento un indirizzo url, che viene caricato su una pagina del browser quando essa viene chiamata.
- WebDriverWait(driver,s).until.(EC.element_to_be_clickable(By.XPATH,Xpath"))→ Questa funzione aspetta s secondi finché l'elemento contrassegnato dal Xpath corrispondente risulta cliccabile. Questa funzione è molto utile per prevenire problemi dove l'elemento viene localizzato prima che il bottone sia effettivamente cliccabile, situazione nella quale frequentemente si presentano degli errori.
- WebDriverWait(driver,s).until.(EC. presence_of_element_located (By.XPATH, "Xpath"))→ Questa funzione aspetta s secondi finché l'elemento contrassegnato dal Xpath corrispondente viene localizzato. Anche questa funzione, come quella precedente, è molto utile per prevenire che vengano chiamati elementi della pagina web che ancora non sono stati caricati.
- driver.find_element()→ Questa funzione trova l'elemento corrispondente all'argomento della funzione. Esistono diversi metodi per localizzare un elemento, quello più utilizzato nel codice è By.XPATH dove si cerca l'elemento corrispondente all'indirizzo Xpath.
- driver.execute_script()→ Esegue comandi JavaScript nella pagina utilizzata, nel nostro caso viene utilizzata per eseguire la funzione "window.scrollTo" che permette di eseguire uno scroll, sia lateralmente che verticalmente, all'interno della pagina. Questa funzione viene utilizzata per far scorrere la pagina affinché funzioni come driver.find_element() possano trovare elementi che non erano inizialmente visibili e generando quindi un errore.
- webdriver.ActionChains()→ Le actionchains sono un metodo per automatizzare interazioni di "basso livello" come lo spostamento del cursore, la pressione di un tasto o l'interazione con un menù. Il nome action chains deriva dal modo in cui vengono eseguite, sono infatti una catena di azioni che si succedono quando vengono chiamate dalla funzione. perform(). Nel specifico di questo progetto vengono utilizzate per spostare il mouse fuori dalla finestra del menu superiore, il quale viene aperto quando si effettua uno scroll verso l'alto. In questo modo si chiude il menù, il quale impediva di selezionare gli elementi sottostanti.
- .sendkeys()→ Questa funzione permette di inviare un input di tasti che possono essere sia lettere della tastiera ma anche tasti funzione specifici come le frecce direzionali o nel caso del progetto il tasto ENTER.
- .click() → Questa funzione rappresenta uno dei modi più elementari con il quale il tool webdriver interagisce con la pagina web. Utilizzando la funzione.click() si esegue infatti una pressione del tasto sinistro del mouse.

4 Workflow del codice

Come precedentemente anticipato, il vantaggio di utilizzare Selenium come web scraper è la possibilità non solo di scaricare i dati ma anche di interagire, emulando un comportamento umano, con il web browser. Di questa ultima caratteristica fa utilizzo il codice per navigare fra le varie tipologie di borse.

Una volta che il codice ha aperto (funzione driver.get) la pagina principale di Farfetch, esso naviga tramite la finestra a scomparsa tra le varie tipologie di borsa, raccolte nella colonna di destra.

Una volta cliccata la tipologia (il programma sceglierà in ordine Borse da mare, Borse a secchiello ecc.) si entra nei loop principali del programma. Ci sono 3 loop racchiusi uno dentro l'altro:

- Il primo esegue il codice al suo interno per ogni tipologia di borsa
- Il secondo esegue il codice per ogni pagina di risultati. Considerando che vengono mostrate 90 borse per pagina, se ci sono 200 risultati, ad esempio, ci saranno tre pagine con la terza contenente 20 elementi.
- Il terzo loop esegue il codice per ogni borsa presente nella pagina.

Le condizioni di uscita dei tre cicli for sono:

- Quando viene generato un errore dovuto al fatto che sono finite le borse nella pagina, in quel caso viene aggiunto nell'url la dicitura "?page=x" dove x è il numero di pagina successivo.
- Nel secondo loop si esce solo quando sono finite le pagine per quella tipologia di borsa.
- Una volta tornati al terzo loop, esso si conclude dopo che sono state analizzate tutte le tipologie di borse, viene chiuso il browser (funzione driver.quit) e quindi il programma.

Dopo aver selezionato la tipologia di borsa, si ottiene una pagina dove vengono visualizzate 90 elementi, con 3 borse per riga. Ogni elemento ha un indirizzo Xpath identico, meno il numero identificativo che per l'appunto va da uno a 90. Si può pensare all'Xpath come una via, e il numero identificativo come il civico, incrementando il numero ogni volta, e utilizzando la funzione driver.click, si effettua un click con il tasto sinistro per ogni borsa presente nella pagina.

Una volta all'interno della singola borsa, Il codice eseguito per raccogliere i dati si basa sul concetto che le funzioni driver.get e webdriverwait per estrapolare il dato specifico hanno bisogno che esso sia visualizzato. Vengono quindi implementate funzioni che scorrono la pagina e cliccano nelle sezioni di pagina contenenti i dati. Una volta che essi vengono visualizzati possono essere memorizzati.

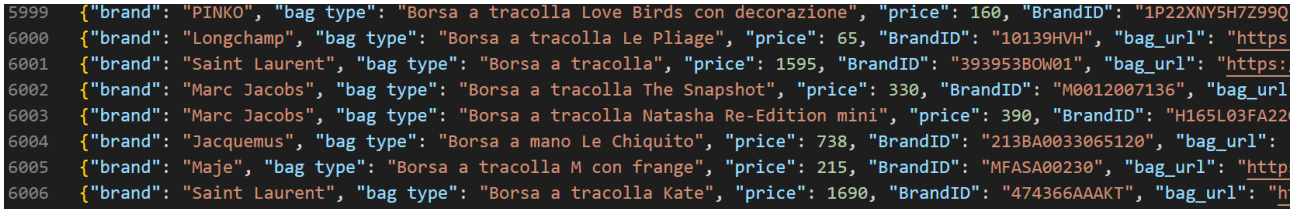
I dati raccolti sono:

- Brand → il marchio della borsa (Gucci, Prada, Yves Saint Laurent…)
- bag_type → la tipologia di borsa, essa può differenziarsi da quella selezionata inizialmente, ad esempio ci possono essere delle borse a secchiello tra le borse da mare
- price → il prezzo, qualora ci fosse uno sconto verrebbe stampato il prezzo scontato
- png → un indirizzo url contenente una immagine della borsa

- descrizione → una descrizione data dal venditore della borsa
- materiali → i materiali presenti all'interno con relative percentuali-
- BrandID → un codice identificativo della borsa
- Dimensioni → vengono salvate in un array misure come la larghezza, altezza, lunghezza della tracolla o del manico
- Bag_url →l'indirizzo url della borsa
- Data e ora → l'ora e la data quando la borsa è stata scansionata

Ci sono alcuni elementi il cui indirizzo specifico varia da borsa a borsa, per questo motivo alcuni dati come le dimensioni o i materiali, fanno utilizzo della funzione try except. Quindi quando il programma cerca dei dati ad un indirizzo inesistente, invece di generare un errore, passa all'indirizzo racchiuso nella funzione except, oppure salta direttamente il dato, in quanto potrebbe essere inesistente.

Dopo ogni singola borsa i dati vengono salvati in un dizionario, questo dizionario viene poi inserito nel file Json dove ogni riga rappresenta un dizionario contenente i dati di ogni singola borsa.



9 Porzione del file Json contenente i dati delle borse

Le informazioni contenute nel file Json possono essere sfruttate per ricavare informazioni utili riguardanti il mercato delle borse di lusso offerte sul sito farfetch.com. Nel caso di questo progetto i dati sono stati elaborati utilizzando Pandas, una libreria di Python descritto al paragrafo 3.2.2, generando delle tabelle che mostrano graficamente le informazioni contenute all'interno dei dati raccolti.

4.1 Sfide incontrate e superamento

4.1.1 Cookies and ads

Per emulare la pressione del tasto sinistro del mouse e quindi cliccare su elemento, ad esempio per premere sull'icona della borsa, si usa la funzione element.click() dove element è l'elemento da premere.

Affinché questa funzione non generi errori, il bottone o link che viene premuto deve essere visibile, questa condizione non era resa possibile in quanto ci sono dei pop-up che appiano a schermo coprendo l'elemento da cliccare.

Per risolvere questo problema bisogna utilizzare la stessa funzione element.click() sull'elemento del pop-up responsabile per chiuderla. Quando il pop-up dei cookies appariva a schermo, il bottone accetta solo cookies necessari" veniva identificato e premuto. Stessa cosa quando appariva il pop-up della newsletter, dove invece il bottone "X" veniva sempre identificato e premuto

4.1.2 Utilità del try except

I blocchi try except sono utilizzati per gestire gli errori e sono stati molto utili per il funzionamento del progetto. Per ricavare i dati da un elemento è essenziale conoscerne l'indirizzo Xpath corretto, tuttavia, essendo il sito ovviamente pensato per essere usufruito da un umano e non da un programma, il posizionamento e la quantità dei dati non è costante. Per un essere umano, infatti, che le dimensioni di una borsa siano scritte a destra o sinistra, è indifferente. Tuttavia, un codice se non trova le informazioni nell'indirizzo Xpath che gli è stato dato, esso genera un errore poiché non riesce ad adattarsi autonomamente. Grazie a try except il codice può invece gestire l'eccezione e proseguire nell'esecuzione: come suggerisce il nome può infatti provare (try) in un indirizzo e qualora esso generasse un errore, il codice cercherebbe nell'indirizzo racchiuso in except. Questa è solo un esempio dell'utilizzo dei blocchi try except, specifico di questo programma, tuttavia, essi vengono utilizzati in ogni situazione dove deve gestire un errore.

4.1.3 Utilità del webdriver wait

Durante i vari test che sono stati fatti per verificare la correttezza del programma diversi errori sono stati generati da elementi che non erano stati correttamente individuati, tal volta anche da elementi che non avevano generato errori nelle prove precedenti. Per risolvere questa situazione si è fatto ricorso alla funzione webdriverWait.

La funzione webdriver wait ha la peculiarità di aspettare x secondi che la condizione necessaria impostata dall'utente sia verificata. In questo modo si evita che l'elemento cercato venga individuato ma, se ad esempio la pagina non è stata caricata, esso non sia visibile. Questa funzione è estremamente utile, ad esempio, quando è necessario cliccare su un elemento. In questo caso viene utilizzata la condizione "element_to_be_clickable" la quale aspetta che l'elemento individuato sia cliccabile. Ciò evita che l'elemento, il quale potrebbe essere stato correttamente individuato, ma per alcuni motivi non è ancora cliccabile; generando quindi un errore quando la funzione element.click() viene chiamata.

4.1.4 Cambio pagina

Per conoscere quando sia necessario cambiare passare alla tipologia di borsa successiva è necessario conoscere il numero totale di pagine per ogni tipologia. Questo dato è scritto in fondo alle 90 borse rendendolo inaccessibile dalla pagina principale.

Tuttavia, grazie alle funzioni di Selenium che imitano il comportamento umano è possibile utilizzare delle funzioni che eseguono uno scroll nella pagina, portandoci quindi alla fine del documento dove questo dato è reperibile. La stessa funzione viene utilizzata anche quando, dopo aver fatto lo scroll verso l'alto, la finestra superiore viene aperta, coprendo le icone delle borse e impedendo il corretto funzionamento del programma (errore element_not_clickable). Utilizzando le funzioni "actionchains"(paragrafo 3.4) contenute nella libreria "selenium.webdriver" è possibile chiamare una funzione che emula lo spostamento del mouse verso il basso chiudendo la finestra prima che la funzione che preme su ogni elemento venisse chiamata, evitando quindi l'errore.

Come scritto in precedenza ci sono 90 elementi per pagina, una volta finiti gli elementi è necessario passare alla pagina successiva, per fare ciò è stato sperimentato di cambiare l'URL o di premere sul bottone "successiva". Alla fine, è stato optato per la prima opzione, in quanto la seconda, pur essendo più intuitiva presentava lo svantaggio che una volta arrivati all'ultima pagina sarebbe stato presente un bottone non cliccabile, il quale generava un errore. Al contrario con il primo metodo una volta arrivati all'ultimo elemento, il programma controlla di non essere all'ultima pagina e modifica l'indirizzo url con quella pagina successiva.

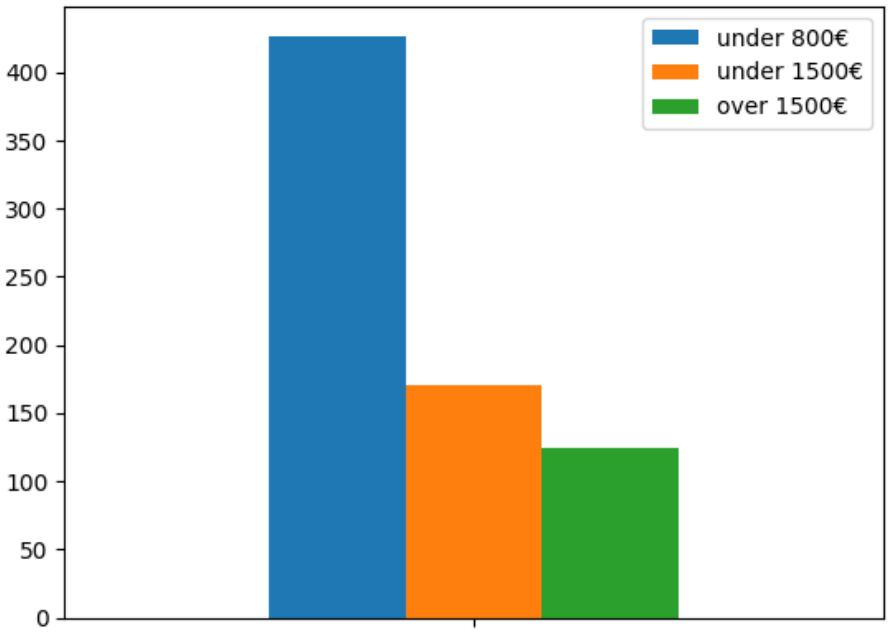
4.1.5 Problema selezione tipologia borsa

All'inizio del progetto per muoversi tra le varie tipologie di borse, si era pensato di scrivere nella barra di ricerca la tipologia di borsa per poi premere cerca. Dopo aver lavorato a lungo con questa modalità ci si è accorti che la ricerca non dava risultati precisi, bensì la ricerca racchiudeva un gran numero di elementi che con il susseguirsi delle pagine erano sempre meno coerenti con la tipologia cercata. Ad esempio, ricercando nella barra di ricerca "borsa a tracolla", la ricerca produceva risultati per 337 pagine, contenendo i prodotti di diverse tipologie, dai costumi da bagno alle scarpe.

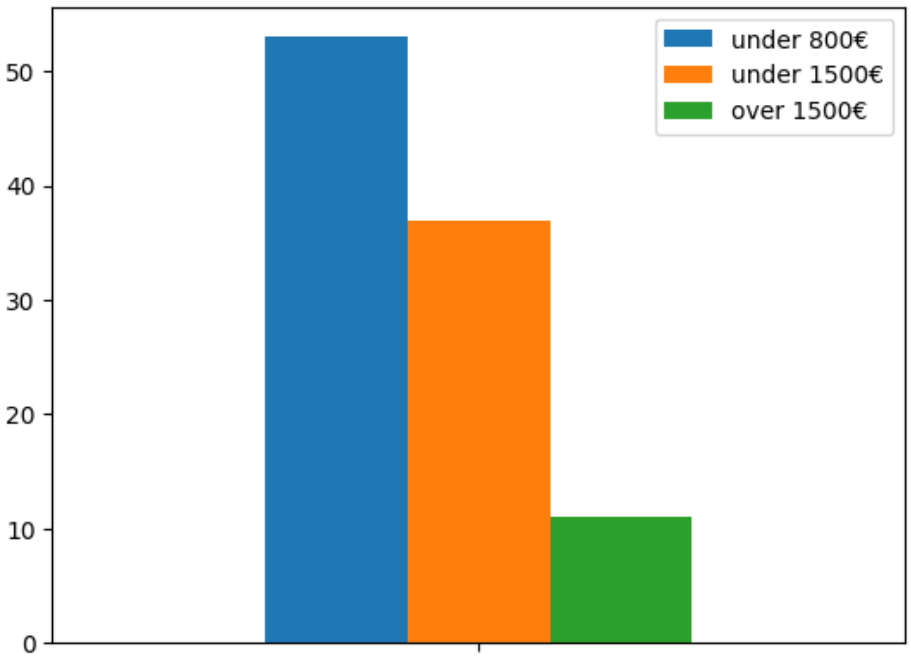
5 Risultati e discussioni

Come è stato anticipato precedentemente, i dati sono stati elaborati utilizzando Pandas per poi generare delle tabelle che mostrano graficamente la distribuzione delle borse su Farfetch.com.

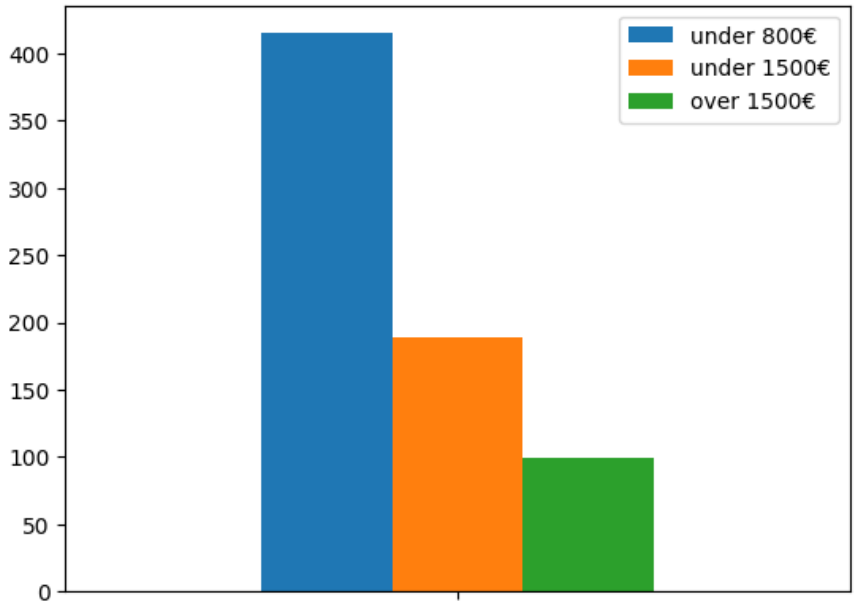
In questo progetto i parametri su cui ci si è concentrati maggiormente sono il prezzo e la tipologia di borsa. Sono i due dati che incidono maggiormente nell'offrire una visione di insieme alla distribuzione delle borse sul e-tailer. Nei primi istogrammi il pool di borse viene suddiviso in tre sottocategorie: borse sotto gli 800€, borse tra gli 800€ e i 1500€ e borse sopra i 1500€. Per ogni tipologia di borsa vengono quindi confrontati il numero di borse presenti per le tre sottocategorie. In questo modo è possibile dedurre in quale tipologia di borsa sono più comuni borse più o meno economiche.



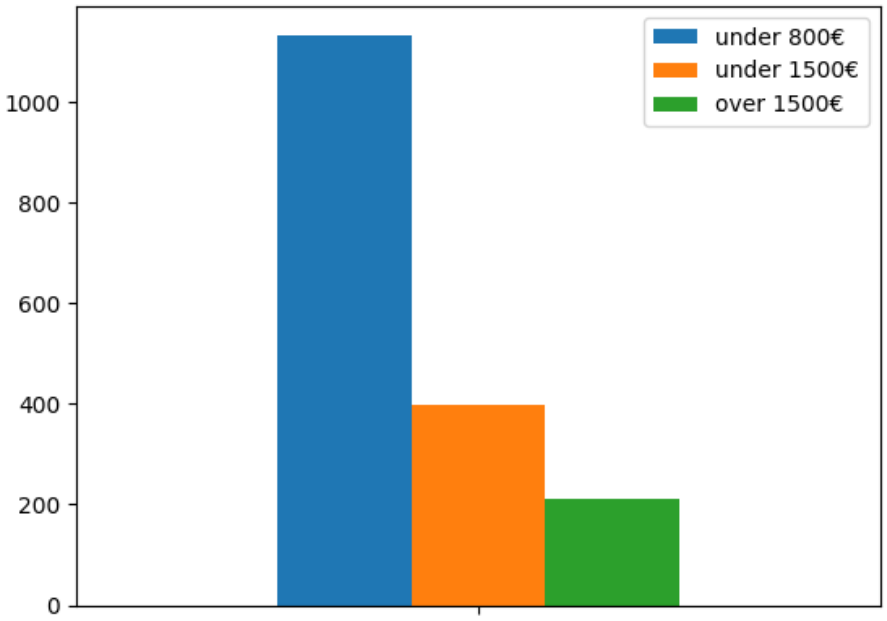
	under 800€	under 1500€	over 1500€
borse a tracolla	427	171	124



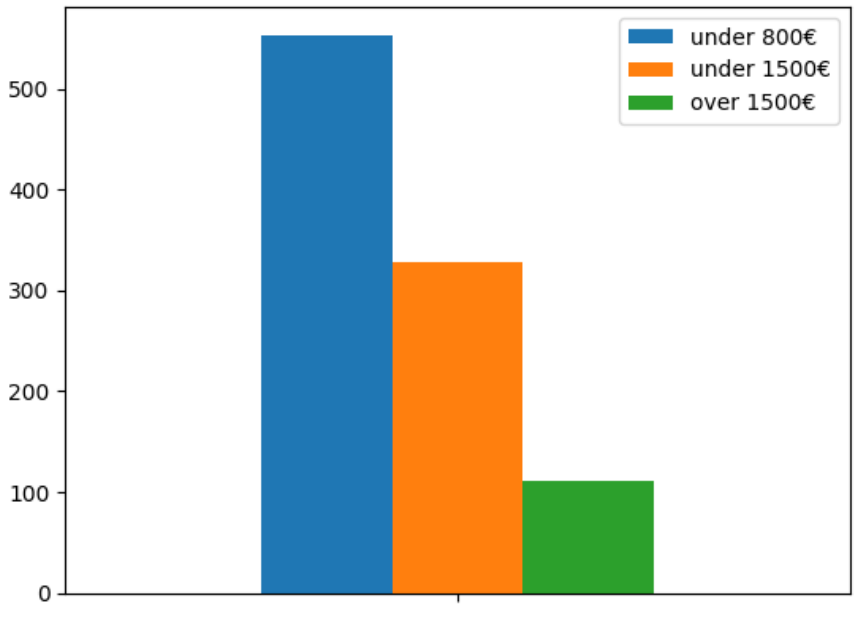
	under 800€	under 1500€	over 1500€
borse da mare	53	37	11



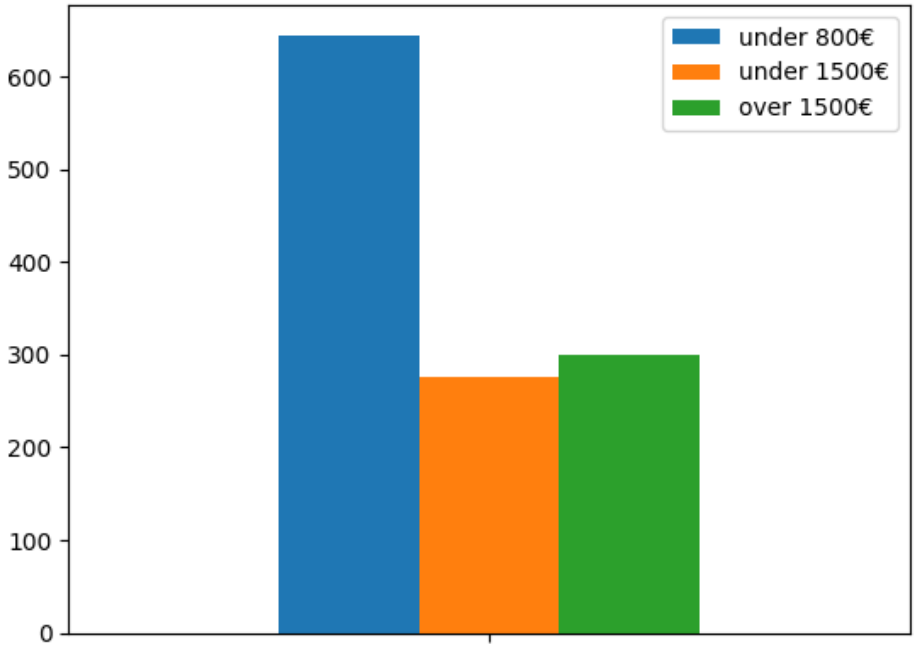
	under 800€	under 1500€	over 1500€
borse a secchiello	415	189	99



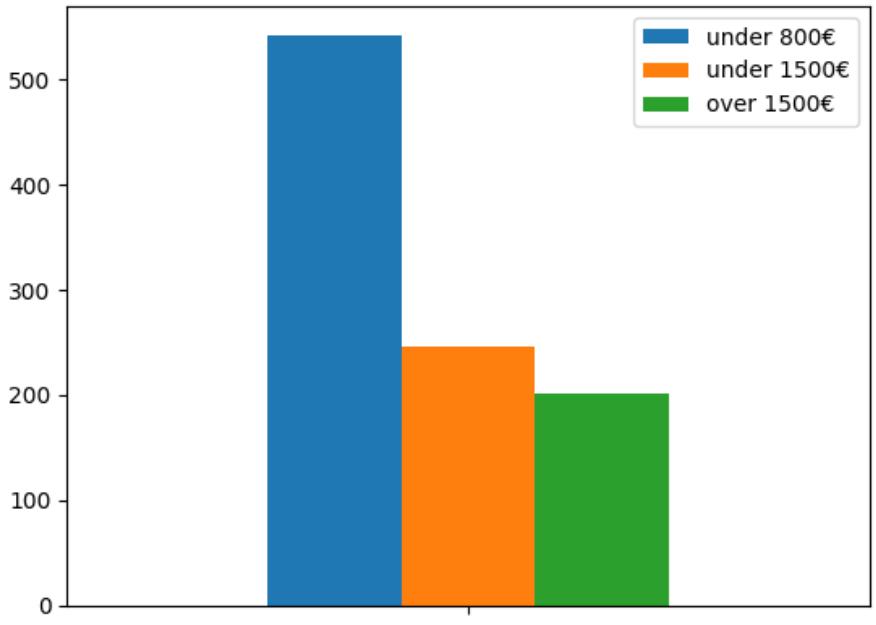
	under 800€	under 1500€	over 1500€
borse clutch	1134	397	211



	under 800€	under 1500€	over 1500€
borse mini	554	328	111



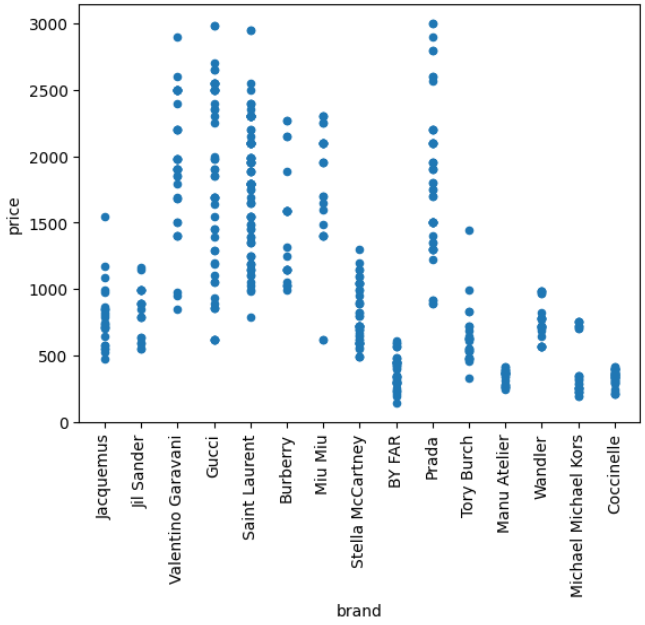
	under 800€	under 1500€	over 1500€
borse a spalla	645	276	299



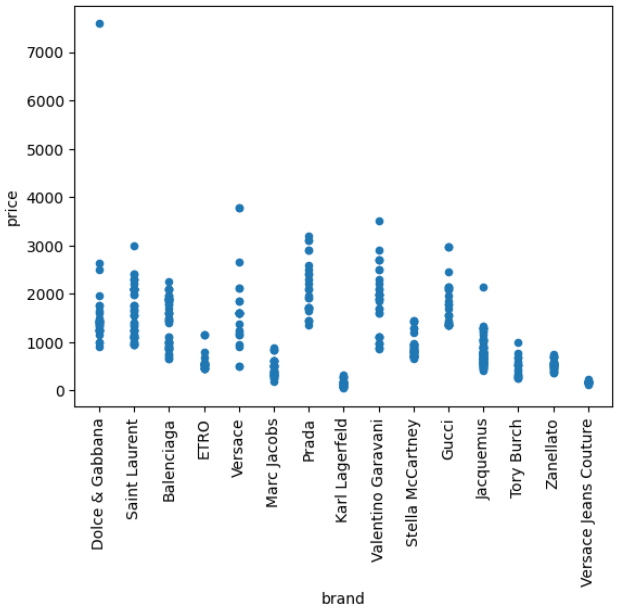
	under 800€	under 1500€	over 1500€
borse tote	543	246	201

All’interno della stessa tipologia, può essere importante, se si vuole avere una visione più completa del mercato, sapere in quale range di prezzo ogni brand opera. Per dimostrare graficamente questi dati si è preferito utilizzare dei grafici a dispersione dove ogni punto rappresenta una borsa, le borse dello stesso marchio si trovano sulla stessa colonna.

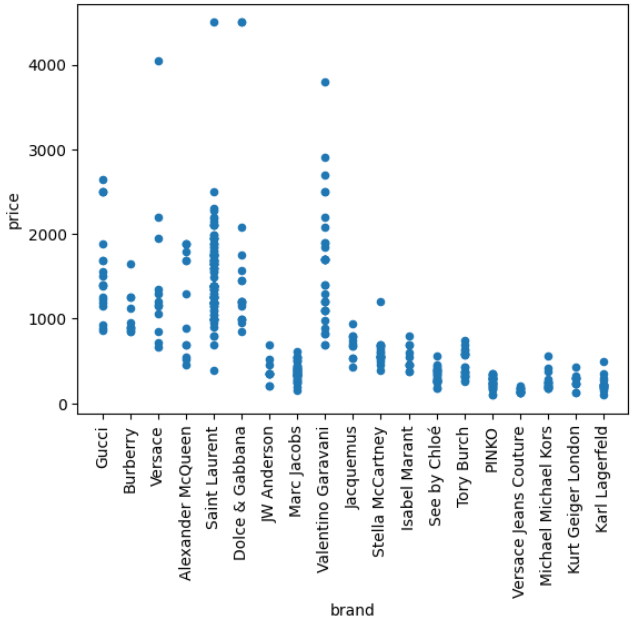
Confronto Prezzo/Brand - Borse a spalla



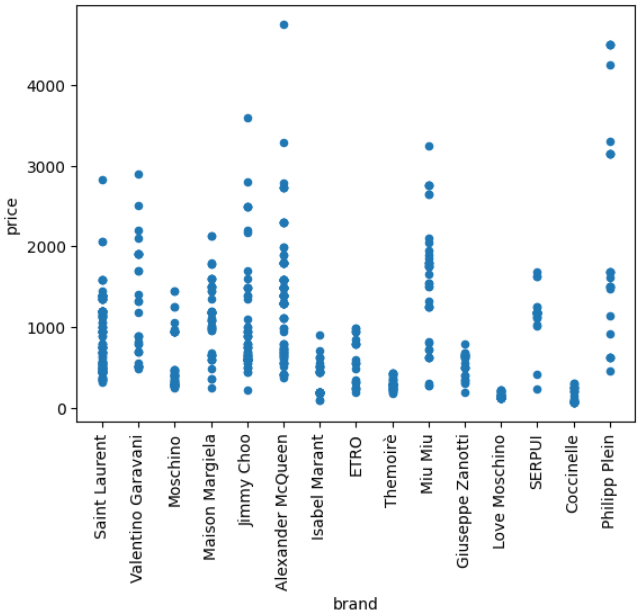
Confronto Prezzo/Brand - Borse tote



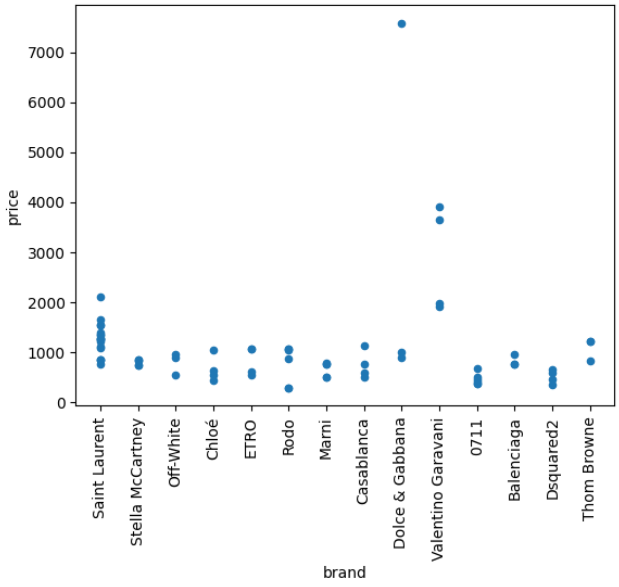
Confronto Prezzo/Brand - Borse a tracolla



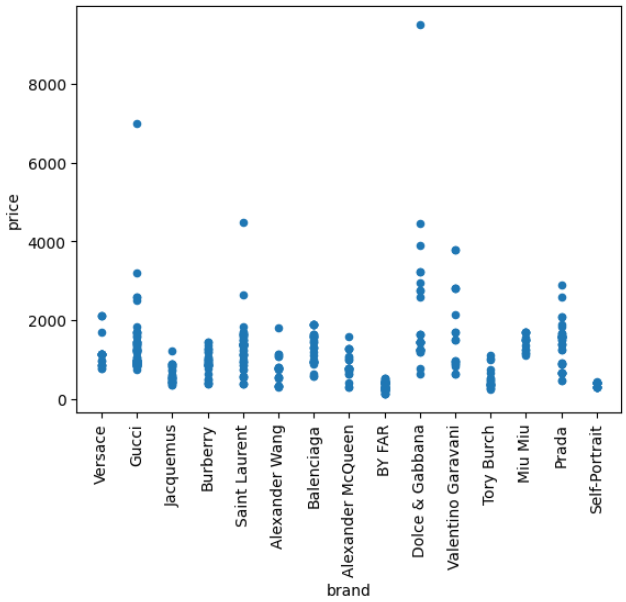
Confronto Prezzo/Brand - Borse Clutch



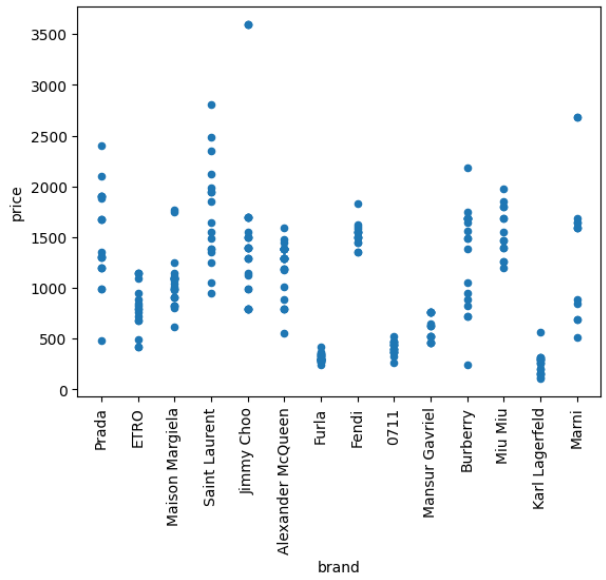
Confronto Prezzo/Brand - Borse da mare

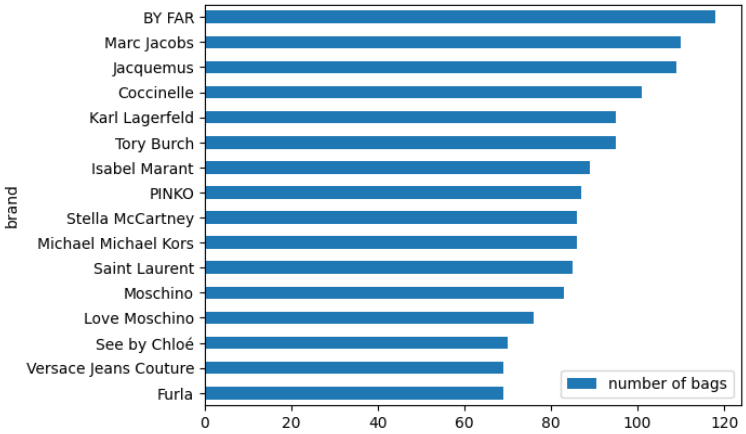


Confronto Prezzo/Brand - Borse mini

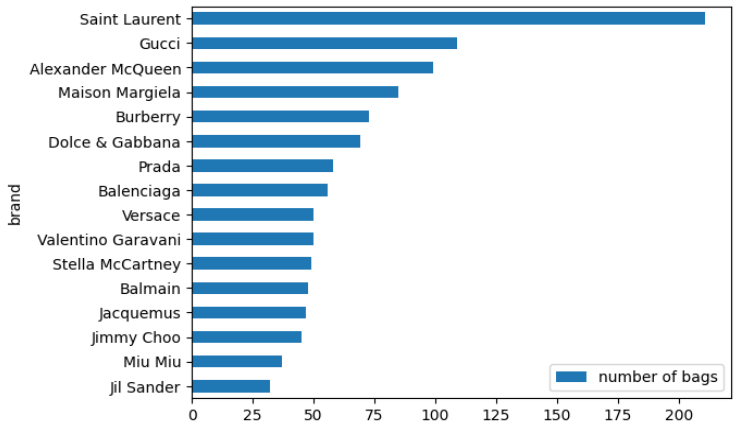


Confronto Prezzo/Brand - Borse a secchiello

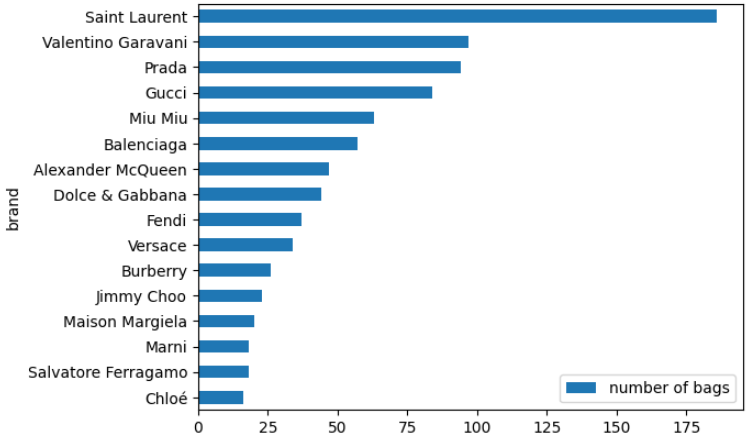




Numero di borse per marchio - sotto gli 800€

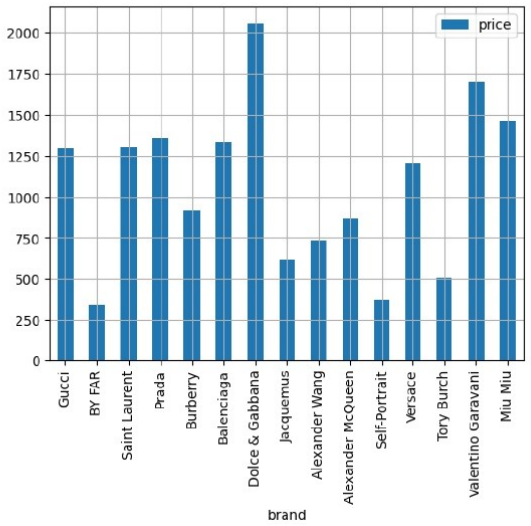


Numero di borse per marchio - sotto i 1500€

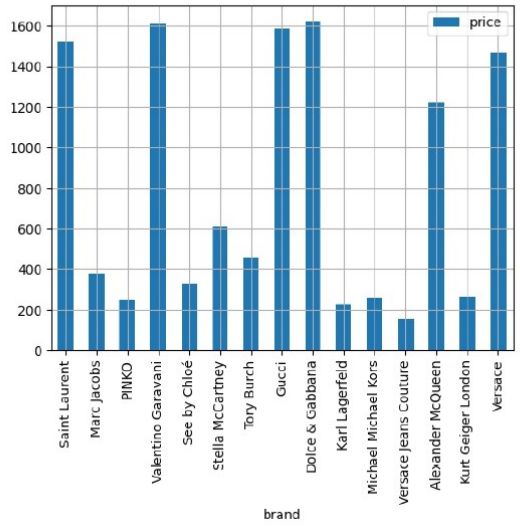


Numero di Borse per marchio sopra i 1500€

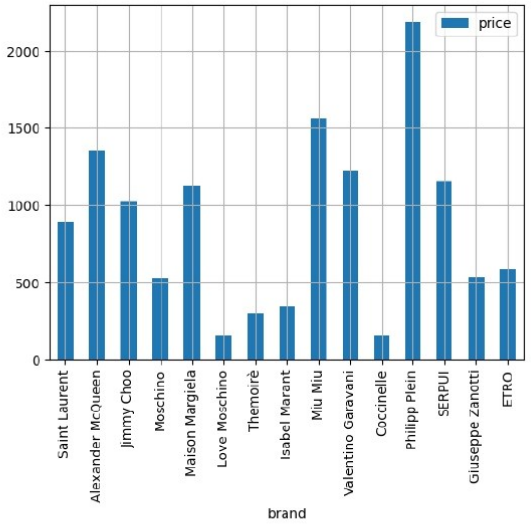
Un altro metodo per visualizzare il range di prezzi su cui un marchio opera, sono stati realizzati degli istogrammi dove sull'asse delle ascisse è presente il marchio mentre su quello delle ordinate il prezzo medio. In questo modo è possibile, non solo confrontare il prezzo medio tra i diversi brand, ma anche il prezzo medio dello stesso brand tra tipologie diverse.



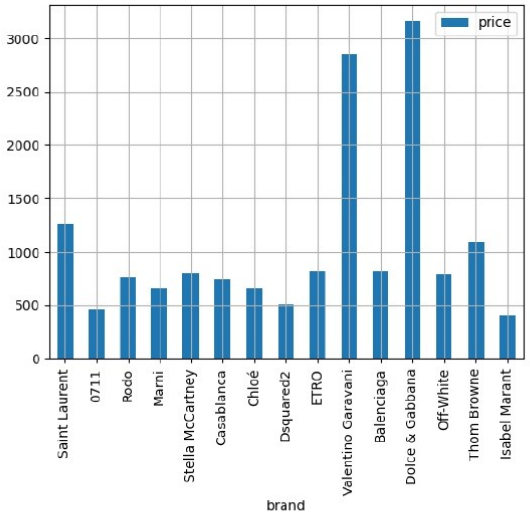
Prezzo medio per brand - Borse mini



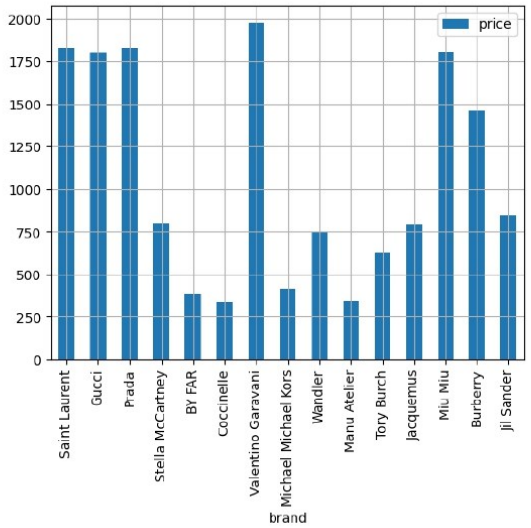
Prezzo medio per brand - Borse a tracolla



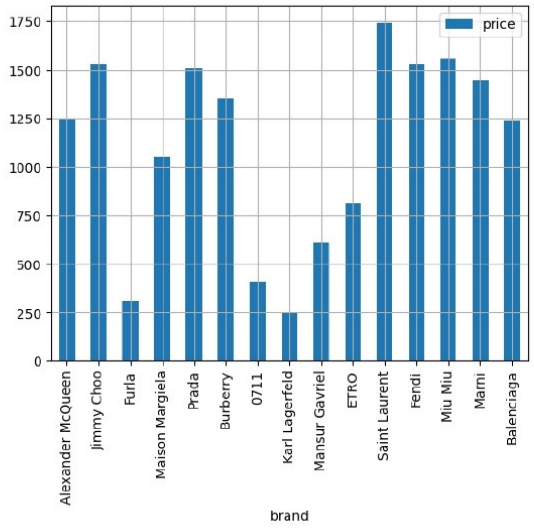
Prezzo medio per brand - Borse Clutch



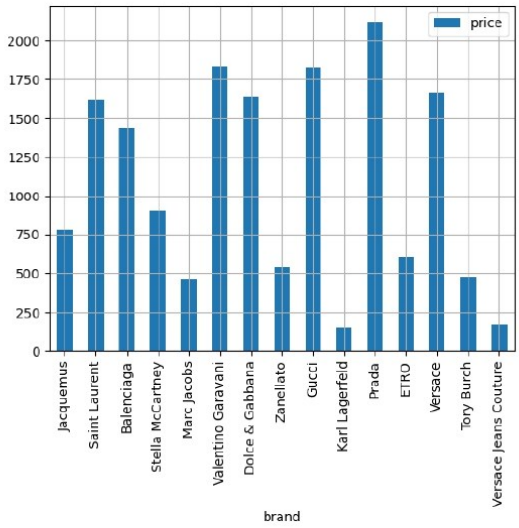
Prezzo medio per brand - Borse da mare



Prezzo medio per brand - Borse Spalla



Prezzo medio per brand - Borse Secchiello



Prezzo medio per brand - Borse Tote

Dai grafici generati dall'estrapolazione dei dati è possibile ricavare uno spaccato del mercato delle borse in vendita su farfetch.com. Come era prevedibile le borse più popolari sono quelle sotto gli 800€, seguite da quelle sotto i 1500€ e infine quelle sopra i 1500€. Tuttavia, è possibile fare delle distinzioni a seconda della tipologia di borsa:

- Le borse da mare e mini hanno un andamento simile, dove sono poco presenti borse sopra i 1500€ mentre quelle mid-range (800-1500€) sono in percentuale superiori alla media.
- Le borse a secchiello, le clutch e le tote hanno un andamento simile.
- Le borse a spalla, è presente una percentuale molto più alta, rispetto alle altre tipologie, delle borse di lusso (sopra 1500€)

Sotto gli 800€ i brand più popolari sono BY FAR, Marc Jacob, Jacquemus e Coccinelle. Questi marchi sono molto popolari in quella fascia di prezzo ma sono quasi non presenti nei range superiori. Al contrario, brand come Saint Laurent, avendo prodotti che si collocano in un range di prezzo molto ampio, è tra il brand più popolari in tutte le fasce di prezzo (11° sotto 800€, 1° sotto i 1500€, 1° sopra i 1500€).

Ci sono altri brand al contrario, che sono praticamente assenti nella fascia di prezzo più economica ma sono molto diffusi nelle altre categorie (Valentino Garavani, Prada, Gucci…), ciò dimostra che pur producendo borse della stessa tipologia, i vari brand puntano a mercati diversi.

6 Conclusione e Sviluppi Futuri

In questo progetto è stato costruito un web scraper per raccogliere dati dal sito farfetch.com relativi ad ogni borsa presente sul sito per sette diverse tipologie, per un totale di 6471 elementi. Il web scraper è stato progettato utilizzando la tecnica mimicry dove il programma emula il comportamento di utente umano per visualizzare i dati di ogni borsa presente. I dati estrapolati sono stati utilizzati per mostrare uno spaccato della distribuzione delle borse sia per tipologia che per prezzo. Le informazioni ricavabili da questo lavoro possono essere utili in diversi contesti. Ad esempio, si possono confrontare due diversi e-tailer per vederne le differenze e i punti di forza, questo è molto utile se si è un potenziale investitore o se si vuole comprendere meglio la concorrenza. I gestori di Farfetch stesso possono capire meglio in quale settore investire e quali sono più saturi. Le aziende produttrici di borse possono utilizzare queste informazioni per comprendere meglio il loro posizionamento del mercato all'interno dell'e-tailer. Queste sono solo degli esempi dell'utilizzo di un web scraper di questa tipologia. Si potrebbe ad esempio ampliare la quantità di informazioni generate, facendo un'analisi degli articoli in vendita su Farfetch in due momenti diversi, per poi analizzare le differenze tra i dati ed estrapolare informazioni sulle tendenze di mercato.

Questa tecnica di estrapolazione dati è estremamente utile per tutti i siti di e-commerce per qualunque tipologia di prodotti poiché ogni settore potrebbe giovare di informazioni e insight analoghi a quelli ricavati in questo contesto. Quanto fatto può inoltre essere allargato al di fuori del settore dell'e-commerce ed essere utilizzato per ricavare informazioni in tutti i settori dove si utilizzano dati storici per fare previsioni. Alcuni esempi sono il settore del marketing, le previsioni finanziarie e gli studi di carattere scientifico come ad esempio la meteorologia.

Fonti bibliografiche

[1]. Hair Jr., J., Page, M., & Brunsveld, “Essentials of Business Research Methods (4th ed.)”, Routledge. 2019.

[2]. Steve LaValle, M. S. H. Analytics: The new path to value. MIT Sloan Management Review. 24 October 2010

[3]P. Wazurkar, R. S. Bhadoria and D. Bajpai, "Predictive analytics in data science for business intelligence solutions," 2017 7th International Conference on Communication Systems and Network Technologies (CSNT), 2017.

[4]. Zhao, Bo. “Web Scraping”. doi: 10.1007/978-3-319-32001-4_483-1, 2017

[5.] R. Diouf, E. N. Sarr, O. Sall, B. Birregah, M. Bousso and S. N. Mbaye, "Web Scraping: State-of-the-Art and Areas of Application," 2019 IEEE International Conference on Big Data (Big Data), 2019.

[6]. R. Diouf, E. N. Sarr, O. Sall, B. Birregah, M. Bousso and S. N. Mbaye, "Web Scraping: State-of-the-Art and Areas of Application," 2019 IEEE International Conference on Big Data (Big Data), 2019

[7]. Stack overflow developer survey 2019. Stack Overflow. (n.d.) Retrieved from https://insights.stackoverflow.com/survey/2019#most-popular-technologies

[8]. Gunawan, R., Rahmatulloh, A., Darmawan, “Comparison of web scraping techniques: Regular expression, HTML DOM and Xpath.”, 2019.