



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

**Web Scraping e Tennis: estrazione,  
salvataggio ed analisi di dati statistici dal  
sito dell'ATP**

*Relatore:*

PROF. DI NUNZIO GIORGIO MARIA

*Laureando:*

GHIOTTO ANDREA

1216363

Anno Accademico 2022/2023



## **Abstract**

Internet è, ad oggi, la più grande raccolta di dati esistente, ed il web diventa così sempre più la principale fonte dalla quale ricavare informazioni. Estrarre dati dal web, spesso, risulta però un'operazione troppo dispendiosa; il pratico copia-incolla non può essere infatti una strada percorribile con una mole di dati nell'ordine dei milioni ed il tempo e le energie impiegate sarebbero enormi.

Fortunatamente si collocano in questo ambito delle tecniche che permettono di rendere più fruibile e meno dispendioso il processo di estrazione di dati ed informazioni dai siti web. Il Web Scraping, argomento principale di questo elaborato, è, infatti, un insieme di tecniche informatiche mediante le quali vengono estratti dati dal web.

In questa tesi si vuole introdurre, presentare e descrivere questo argomento in tutte le sue sfumature, citandone le origini, mostrando come approcciarvisi, descrivendone strumenti, tecniche e librerie. Viene poi presentato e descritto il progetto che è stato sviluppato, con il quale si è applicato il Web Scraping nel campo delle analisi di dati statistici in ambito sportivo, nello specifico quello tennistico.



# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Che cos'è il Web Scraping?</b>	<b>3</b>
1.1 Un po' di storia	3
1.1.1 Il Worl Wide Web	3
1.1.2 Il Wanderer	3
1.1.3 BeautifulSoup	4
1.1.4 Presente e futuro	4
1.2 Web Crawling e Web Scraping	4
1.2.1 Web Crawling	4
1.2.2 Web Scraping	6
1.2.3 Questioni legali	7
<b>2 Strumenti, tecniche e librerie per il Web Scraping</b>	<b>9</b>
2.1 Requests e BeautifulSoup	9
2.2 Selenium	10
2.3 Librerie per la raccolta dati	11
2.3.1 Pandas	11
2.3.2 Numpy	11
2.3.3 Matplotlib e Seaborn	11
<b>3 Dati statistici nello sport: il tennis e l'ATP</b>	<b>13</b>
3.1 La statistica nel mondo dello sport	13
3.1.1 Le origini e l'approccio di Beane nel baseball	13
3.1.2 Dati e statistica nello sport moderno	14
3.2 Il tennis	15
3.2.1 L'ATP e atptour.com	15
<b>4 Sviluppo dello scraper</b>	<b>17</b>
4.1 Obiettivi	17
4.1.1 Il servizio e le diverse superfici	17
4.1.2 L'evoluzione del servizio nel corso degli anni	18
4.2 Approccio iniziale	18
4.2.1 Dati da estrarre	18

4.2.2	Struttura del sito . . . . .	19
4.2.3	Librerie e tool scelti . . . . .	21
4.2.4	Idea di partenza . . . . .	21
4.3	Sviluppo programma . . . . .	21
4.3.1	Accesso al sito . . . . .	21
4.3.2	Estrazione dei dati . . . . .	23
4.3.3	Salvataggio dei dati . . . . .	23
4.3.4	Output . . . . .	25
<b>5</b>	<b>Analisi dei dati estratti . . . . .</b>	<b>27</b>
5.1	Il servizio e le diverse superfici . . . . .	27
5.2	L'evoluzione del servizio nel corso degli anni . . . . .	28
<b>6</b>	<b>Conclusioni . . . . .</b>	<b>31</b>
	<b>Bibliografia . . . . .</b>	<b>33</b>

## Elenco delle figure

1.1	Il web . . . . .	5
1.2	Schema riassuntivo Web Crawling . . . . .	5
1.3	Schema sintesi Web Scraping . . . . .	6
1.4	Comunicazione tra Client e Server . . . . .	7
2.1	Requests e BeautifulSoup . . . . .	10
2.2	Logo Selenium . . . . .	10
3.1	Billy Beane . . . . .	14
3.2	Il Centre Court di Wimbledon: il più famoso stadio di tennis del mondo . .	15
3.3	Logo ATP . . . . .	15
3.4	Home page di atptour.com . . . . .	16
4.1	Parte della tabella con i dati del servizio relativi al 2022 e a tutte le superfici	19
4.2	Parte dello script che genera le tabelle . . . . .	20
4.3	Codice HTML del body delle tabelle . . . . .	20
4.4	Link alla tabella con i dati del servizio relativi al 2022 e a tutte le superfici	20
4.5	Import ed url . . . . .	22
4.6	Struttura iterativa del codice, creazione driver e tabelle . . . . .	22
4.7	Ricerca ed estrazione dei dati . . . . .	23
4.8	Salvataggio dati . . . . .	24

---

4.9	Elif per diverse superfici . . . . .	24
4.10	Creazione tabelle . . . . .	25
4.11	Tabelle excel anno per anno e superficie per superficie . . . . .	25
4.12	Cartella finale con programma Python e tabelle . . . . .	25
4.13	Parte della tabella relativa ai dati del 2022 e della superficie hard . . . . .	26
4.14	Tabella Serve Rating . . . . .	26
5.1	Grafico valori medi del Serve Rating nelle diverse superfici . . . . .	27
5.2	Grafico valori medi dell’Avg. Aces/Match nelle diverse superfici . . . . .	28
5.3	Grafico andamento Serve Rating negli anni . . . . .	29
5.4	Grafico andamento Avg. Aces/Match negli anni . . . . .	29





# Introduzione

Nel mondo dello sport moderno l'analisi dei dati e la statistica hanno assunto un ruolo sempre più rilevante. La ricerca, l'estrazione ed il salvataggio dei dati relativi agli atleti, delle statistiche riguardanti vari aspetti di molteplici discipline o delle informazioni riguardanti le performances degli avversari, sono quindi operazioni che appaiono essere sempre più decisive per orientare decisioni e strategie.

Il Web Scraping risulta quindi essere uno strumento essenziale per automatizzare l'estrazione ed il salvataggio dei dati che si vogliono poi analizzare.

Il progetto che si andrà a sviluppare all'interno di questa tesi è uno scraper, ovvero un programma che applica la tecnica del Web Scraping, per l'estrazione ed il salvataggio di dati dal sito dell'Association of Tennis Professionals (ATP), [atptour.com](http://atptour.com).

I dati di interesse saranno alcune statistiche legate al servizio, colpo di inizio di un punto nel tennis, con l'obiettivo di, in fase di analisi, poter discutere e provare alcuni concetti come la maggior efficienza del servizio sulle superfici veloci e la sua evoluzione nel corso degli anni.

L'elaborato è strutturato nel seguente modo:

- Capitolo 1: introduzione al Web Scraping con cenni storici e nozioni teoriche
- Capitolo 2: descrizione dei vari strumenti, tecniche e librerie che si possono utilizzare per fare Web Scraping
- Capitolo 3: introduzione al mondo della statistica sportiva e descrizione dell'ambito, quello tennistico, nel quale si opererà
- Capitolo 4: presentazione e descrizione del progetto realizzato, dagli obiettivi iniziali alla sua realizzazione
- Capitolo 5: analisi dei dati estratti
- Capitolo 6: conclusioni e considerazioni finali



# Capitolo 1

## Che cos'è il Web Scraping?

### 1.1 Un po' di storia

Sebbene possa sembrare un concetto recente, il Web Scraping [1] esiste sin dagli inizi di Internet, e, nonostante si tenda ad associarlo all'estrazione di grandi quantità di informazioni dai siti web, esso venne creato con uno scopo completamente diverso: rendere il World Wide Web più facile da usare.

#### 1.1.1 Il World Wide Web

Le origini delle primissime basi del Web Scraping possono essere fatte risalire al 1989, quando lo scienziato britannico Tim Berners-Lee creò il World Wide Web. L'idea originale era di creare una piattaforma che consentisse a professori e ricercatori universitari nelle università e negli istituti di tutto il mondo di condividere informazioni.

Sebbene fosse molto meno visivo e molto più piccolo dell'Internet di oggi, aveva tre importanti funzionalità che gli strumenti di Web Scraping utilizzano ancora oggi:

- URL: ora utilizzati per designare uno scraper per un sito web specifico.
- collegamenti ipertestuali incorporati: che ci consentono di navigare attraverso il sito web designato.
- pagine web: contenenti vari tipi di dati: testo, immagini, audio, video, ecc.

Continuando il suo lavoro, due anni dopo, Tim Berners-Lee creò il primo browser web, una pagina web `http://`, dando così alle persone un modo per accedere ed interagire con il World Wide Web.

#### 1.1.2 Il Wanderer

Pochi anni dopo, nel 1993, nacque il primo web robot, il World Wide Web Wanderer. Creato da Matthew Gray presso il Massachusetts Institute of Technology, Wanderer era un web crawler, basato su Perl, che misurava le dimensioni del World Wide Web. Anche se

l'autore non lo rivendica, il Wanderer aveva il potenziale per diventare il primo motore di ricerca World Wide Web per uso generale. Tuttavia, sempre nel 1993, è nata la tecnologia che ha gettato le basi per grandi nomi come Google, Bing, Yahoo e altri strumenti di ricerca sul web oggi.

### 1.1.3 BeautifulSoup

Poco più di un decennio dopo, nel 2004, è arrivata BeautifulSoup - HTML parser, una libreria di script ed algoritmi comunemente usati, scritta nel linguaggio di programmazione Python. BeautifulSoup aiutava i programmatori a comprendere la struttura del sito e ad analizzare il contenuto di contenitori HTML, salvandoli da ore di noioso lavoro. Rimane tutt'ora una delle librerie più avanzate e sofisticate per il Web Scraping. Pochi anni dopo l'uscita di BeautifulSoup, è nato quello che può essere definito come il moderno Web Scraping.

### 1.1.4 Presente e futuro

Diverse aziende hanno lanciato piattaforme software di Web Scraping visivo che consentivano agli utenti di evidenziare manualmente le informazioni che desideravano estrarre salvandole in un foglio di calcolo o in un database Excel. Questi programmi avevano interfacce utente semplici, consentendo ai non programmatori di estrarre facilmente i dati dal web. Al giorno d'oggi, con il progresso delle tecnologie e delle industrie, le aziende cercano di ottenere un vantaggio rispetto alla concorrenza. Poiché la quantità di informazioni disponibili su Internet sta crescendo in modo esponenziale, il Web Scraping sta diventando uno dei metodi più importanti e più utilizzati per acquisire dati su larga scala in molteplici settori. Il Web Scraping è cresciuto enormemente negli ultimi anni e la sua crescita è destinata a continuare.

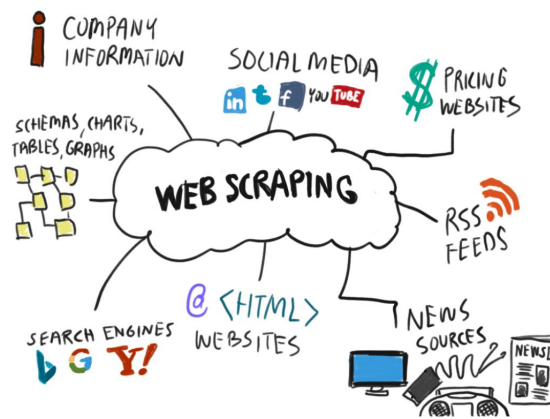
## 1.2 Web Crawling e Web Scraping

Il web è una delle principali fonti di raccolta dati e spesso la loro estrazione da questa fonte può risultare dispendiosa, a causa dell'enorme mole di dati con cui si ha a che fare e dell'eterogeneità delle informazioni disponibili.

Le tecniche di Web Crawling e Web Scraping [2] [3] si collocano in questo contesto con l'obiettivo di rendere più fruibile e meno dispendioso il processo di estrazione di dati ed informazioni dai siti web.

### 1.2.1 Web Crawling

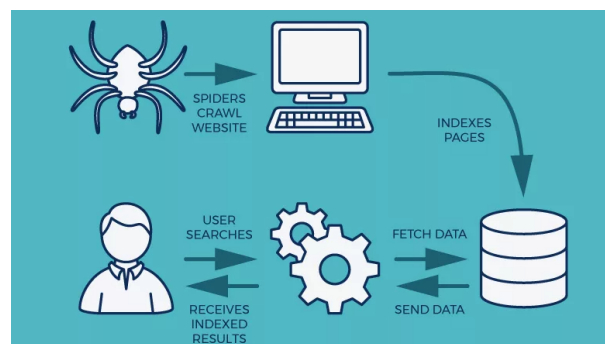
Il Web Crawling è un processo di analisi dei siti web utilizzato per indicizzarne i contenuti, ovvero per fornire al motore di ricerca le corrette informazioni affinché possa



**Figura 1.1:** Il web

catalogare i contenuti dei siti web nel proprio database, restituendoli all'utente attraverso la SERP (Search Engine Page Results), la pagina dei risultati di un motore di ricerca.

Un crawler è quindi un bot Internet che scandaglia il web per crearne una “mappa”; questi bot sono noti anche come “spider”, ovvero “ragni”, proprio perchè si muovono lungo tutta la ragnatela globale del web come dei veri e propri ragni.



**Figura 1.2:** Schema riassuntivo Web Crawling

Un programma che effettua crawling di un sito web generalmente parte da una lista di URL ed esegue una richiesta http (o https) per ogni URL, ottenendo come risposta il contenuto della pagina, dalla quale andrà a ricavare, e ad aggiungere alla lista, tutti i nuovi collegamenti trovati. Per trovare tutte le informazioni pertinenti che Internet offre, ci sono tre diversi percorsi principali che un crawler può seguire:

- cercare collegamenti ipertestuali: lista di URL di siti web, che il programma dovrà visitare sistematicamente. Gli URL di questa lista, chiamata Crawl Frontier (frontiera di indicizzazione), vengono visitati più volte così da poterne registrare eventuali modifiche o aggiornamenti.
- scansionare la sitemap: una sitemap XML è un file in cui vengono fornite informazioni su pagine, video ed altri file importanti presenti sul sito, nonché sulle correlazioni tra i vari elementi. I motori di ricerca come Google leggono questo file tramite i crawler per eseguire una scansione più efficiente.

- invio manuale: invio manuale delle pagine al motore di ricerca. Questa pratica si usa quando si pubblicano nuovi contenuti oppure quando si aggiornano e si vuole ridurre il tempo necessario per far notare l'update al motore di ricerca.

I crawler seguono ogni collegamento trovato su una pagina web e possono restituire il contenuto ad uno scraper. E' quindi possibile che vengano salvate e rese disponibili dai motori di ricerca delle informazioni che dovrebbero restare private o che non dovrebbero essere indicizzate. Per proteggersi i webmaster possono però creare un file robots.txt nella cartella root del sito web, dove specificare le regole che i crawler dovrebbero seguire, ad esempio:

```
Useragent : *
```

```
Disallow : / admin/
```

che comunica ai crawler di non andare ad analizzare la directory admin.

Oltre a delle regole di esclusione (che se non rispettate possono far incappare in degli ostacoli tali da "intrappolare" il crawler) è possibile fornire anche delle mappe del sito (sitemap), generalmente in formato XML, che elencano gerarchicamente tutte le pagine web. Queste mappe sono ampiamente utilizzate non solo per la navigazione dell'utente, ma anche per facilitare l'operazione di crawling.

Uno dei più famosi web crawler è Googlebot, utilizzato da Google per il suo motore di ricerca. Il crawler di Google è identificabile in quanto le richieste contengono "Googlebot" all'interno del campo user-agent e l'indirizzo host contiene "googlebot.com". Google fornisce inoltre degli strumenti ai webmaster per impostare, tra le altre cose, la frequenza di crawling e per generare e validare i file robots.txt e le sitemap.

### 1.2.2 Web Scraping

Il Web Scraping (letteralmente "raschiatura del web") è una tecnica atta ad estrarre informazioni dalle varie pagine dei siti web tramite delle tecniche automatiche. Nonostante infatti l'estrazione di dati da un sito web possa avvenire manualmente, si parla di Web Scraping quando questa procedura è automatizzata mediante l'utilizzo di software che emulano la visita di un utente ad un sito, salvandone poi i dati di interesse in database o in fogli di lavoro appositi.



Figura 1.3: Schema sintesi Web Scraping

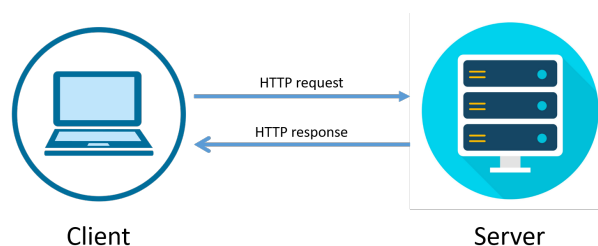
Il Web Scraping è una tecnica ancora più invasiva del Web Crawling, nonostante, spesso, i due tipi di bot vengano confusi. E' importante, infatti, evidenziare una sostanziale differenza: mentre i crawler navigano il web per catalogare le informazioni in modo generalizzato, gli scraper sono programmati per estrarre contenuti da pagine specifiche. Se da un lato i crawler sfruttano anche tecniche di scraping per un'analisi dei contenuti funzionale alla loro indicizzazione, gli scraper memorizzano i dati estratti in database esterni; non a caso, infatti, il Web Scraping è conosciuto anche come Web Harvesting o Web Data Extraction.

Come per i crawler, anche per gli scraper si distinguono diverse modalità d'azione, nello specifico:

- scraping manuale: estrarre e memorizzare volta per volta singole informazioni
- scraping automatico: algoritmi di consultazione ed estrazione dei dati

Ad ogni modo però, l'intero funzionamento di uno scaper può essere diviso in due fasi sequenziali:

- acquisizione delle risorse dal web: un software di Web Scraping inizia inviando, come illustrato in Figura 1.4, una richiesta HTTP ad un sito dal quale si vogliono raccogliere risorse. Il tipo di richiesta può variare: GET (contenente l'URL del sito a cui si vuole accedere ed usato per ottenere il contenuto della risorsa desiderata), HEAD (analogo a GET ma restituisce solo i campi dell'header) e POST (usato quando si vogliono inviare informazioni al server). Una volta che il sito web ha ricevuto con successo la richiesta, invierà al software una risposta che, se correttamente ricevuta (indicato dal codice di stato dell'oggetto `requests.Response()`, uguale a 200), la gestirà nel modo più opportuno.



**Figura 1.4:** Comunicazione tra Client e Server

- estrazione delle informazioni di interesse: superata la fase di acquisizione delle risorse, il software si occupa di gestire i dati grezzi ricevuti, tralasciando i dati di non interesse e strutturando nel modo opportuno quelli di interesse.

### 1.2.3 Questioni legali

Nonostante la frequenza con cui vengono usate le tecniche fin qui descritte, non vi è, ad oggi, una legislazione o un codice etico [4] ben definito che regola l'uso di tali

algoritmi. Questa mancanza può risultare molto scomoda, poiché risulta impossibile per un programmatore sviluppare uno scraper senza essere totalmente sicuro di non incorrere in controversie, etiche o legali, legate al suo lavoro, al suo operato, ai dati estratti o alle informazioni ottenute.

Esempi di alcune controversie legate all'utilizzo di dati ed informazioni prese dai siti web sono: accesso ed uso illegale dei dati, violazione del contratto, uso di materiale protetto da diritto d'autore, appropriazione indebita, violazione del segreto commerciale. Per evitare di incappare in problematiche di questo tipo è bene essere sempre aggiornati ed informati riguardo ai termini e alle condizioni dei siti web a cui si vuole accedere.



# Capitolo 2

## Strumenti, tecniche e librerie per il Web Scraping

Molto spesso, a causa della forma non strutturata del web e delle informazioni eterogenee contenute al suo interno, non è affatto semplice poter reperire ed estrarre i dati di nostro interesse. Qualora infatti i siti web che visitiamo non ci forniscono delle API (Application Programming Interface) per poter accedere più velocemente ai dati, diventa necessario l'uso di strumenti più complessi. Descriviamo ed analizziamo in seguito i più comuni ed i più utilizzati.

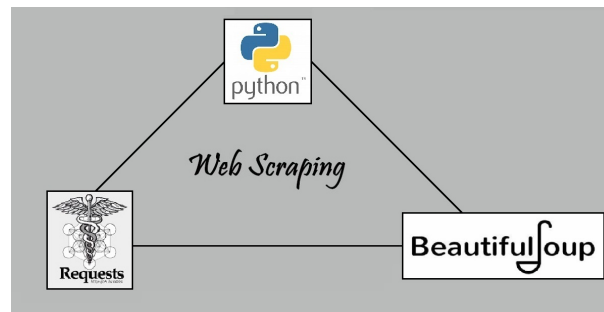
### 2.1 Requests e BeautifulSoup

Requests [5] è una libreria Python che permette di gestire richieste HTTP, stabilendo una comunicazione tra il software ed il sito web. Questa libreria infatti consente di effettuare richieste HTTP e ricevere risposte, sempre HTTP, contenenti la risorsa richiesta (ad esempio il contenuto HTML di una pagina), più altre informazioni come, ad esempio, il codice di stato della risposta.

BeautifulSoup [6] invece è una libreria, sempre del linguaggio Python, usata per il Web Scraping e per leggere dati presenti nelle pagine web o nei documenti HTML; permette infatti di parsificare una pagina HTML, trasformandola in una struttura ad albero, sfruttando quella del DOM (Document Object Model), un'interfaccia di programmazione standardizzata per strutturare documenti HTML ed XML. Per poter operare con questa libreria è necessario creare un oggetto di tipo BeautifulSoup, al quale vanno poi passati il contenuto della pagina HTML ed un parsificatore, il cui compito è quello di rendere i dati leggibili ed estraibili. Tra i vari parsificatori troviamo:

- `html.parser`: parsificatore di default, utilizzato qualora non vengano forniti parametri aggiuntivi. Non richiede installazioni aggiuntive.
- `lxml`: parsificatore molto veloce che richiede però installazioni aggiuntive.

- `html5lib`: più lento dei precedenti, analizza una pagina web analogamente a quanto fa il browser web.



**Figura 2.1:** Requests e BeautifulSoup

L'uso combinato di queste due librerie, Requests per la gestione e l'elaborazione delle richieste e risposte HTTP e BeautifulSoup per il parsing delle pagine HTML, è una soluzione perfetta, e tra le più usate, per effettuare lo scraping di dati dalle pagine e dai siti web.

## 2.2 Selenium

Selenium [7] è il più diffuso tool open source per la gestione automatizzata dei browser. Selenium indica in realtà una suite, composta da diversi strumenti: Selenium IDE, Selenium Builder, Selenium Grid e Selenium WebDriver. Tra questi il componente principale è il Selenium Web Driver, che simula il comportamento di un utente reale all'interno di un browser.



**Figura 2.2:** Logo Selenium

Implementato come un driver per browser, Selenium WebDriver non necessita di un server per eseguire i comandi e, grazie a questo, può interagire con i browser più diffusi, tra cui Chrome, Firefox, Internet Explorer e Microsoft Edge. Ciò però non basta a garantirne un corretto funzionamento: è necessario infatti scaricare ed installare il driver del browser a cui Selenium deve connettersi (ad esempio: ChromeDriver per Chrome, GeckoDriver per Firefox, ecc.). L'interazione tra queste componenti permette poi di ottenere un riscontro visivo e reale dato che, in fase di esecuzione del codice, si aprirà il browser e saranno visibili tutte le azioni da esso simulate, seguendo quanto scritto nel codice dal programmatore.

Una caratteristica molto importante di Selenium, che differenzia questo tool dagli altri, è la gestione molto più semplice delle pagine web dinamiche e dei siti web che fanno un uso massivo di script JavaScript; viene infatti reso possibile un vero e proprio rendering

della pagina e viene eseguito il codice JavaScript, riuscendo quindi a raggiungere i dati popolati in questo modo, cosa che non sarebbe possibile utilizzando altri tool o librerie, ad esempio BeautifulSoup.

## 2.3 Librerie per la raccolta dati

Una volta estratti i dati e le informazioni di nostro interesse attraverso l'utilizzo degli strumenti descritti in precedenza, è necessario poter raccogliere, salvare ed analizzare quanto ottenuto. Per fare ciò è necessario l'utilizzo di librerie specifiche per operazioni di questo tipo. Alcune tra le principali sono qui elencate.

### 2.3.1 Pandas

Pandas [8] è un tool open source di analisi e manipolazione dei dati veloce, potente, flessibile e facile da usare, costruito sulla base del linguaggio di programmazione Python. Si tratta di una libreria utilizzata per la manipolazione e l'analisi di dati; in particolare, offre strutture dati ed operazioni per manipolare tabelle numeriche e serie temporali.

Ad esempio, Pandas DataFrame, è una struttura dati bidimensionale, ovvero i dati sono in allineati in formato tabulare ed è quindi possibile lavorare con righe e colonne. Altra struttura dati interessante di Pandas è Series, utilizzata per rappresentare righe e colonne di un DataFrame.

### 2.3.2 Numpy

Numpy [9] è una libreria open source di Python con funzioni scientifiche aggiuntive, particolarmente utile per eseguire calcoli su vettori e matrici. Aggiunge infatti supporto a strutture dati come array multidimensionali o matrici molto grandi, fornendo una vasta collezione di funzioni matematiche di alto livello per poter operare efficientemente su di esse.

### 2.3.3 Matplotlib e Seaborn

Matplotlib [10] è una delle librerie più utilizzate nel mondo Python. Viene utilizzata per generare visualizzazioni di dati statiche, animate ed interattive, ed ha molte opzioni per la creazione di grafici e personalizzazioni.

Seaborn [11] è invece una libreria di visualizzazione dei dati Python basata su Matplotlib. Fornisce un'interfaccia di alto livello per disegnare grafici statistici e informativi ed è estremamente intuitiva. Infatti, dato un DataFrame Pandas e qualche indicazione sulle righe o colonne da considerare, appare rapidamente tutto ciò che si può evincere da essi.



# Capitolo 3

## Dati statistici nello sport: il tennis e l'ATP

Come citato nell'abstract, in questo progetto di tesi si opererà nell'ambito dei dati statistici nel mondo dello sport [12]; nello specifico il settore di interesse sarà quello tennistico. Verranno infatti estratti dati statistici legati ad alcuni aspetti del gioco, che saranno poi oggetto di analisi sulla base di quanto ottenuto.

E' bene però, prima di procedere con la presentazione e descrizione di quanto prodotto, contestualizzare il mondo della statistica sportiva, il tennis ed il sito dal quale verranno estratti i dati, ovvero quello dell'ATP, [atptour.com](http://atptour.com) [13].

### 3.1 La statistica nel mondo dello sport

Oggi, tutti gli appassionati di sport sanno che i numeri, i dati e le statistiche entrano a pieno titolo nel campo di gioco, dal numero di gol segnati dai calciatori, alle percentuali di canestri realizzati dai cestisti o al numero di basi conquistate dai giocatori di baseball. Non è però sempre stato così, in passato infatti era opinione diffusa che "il caso domina il gioco" e che l'utilizzo della statistica in ambito sportivo fosse follia.

#### 3.1.1 Le origini e l'approccio di Beane nel baseball

Fu proprio il baseball e la figura di Billy Beane, nominato General Manager degli Oakland Athletics nel 1997, a giocare un ruolo centrale nel rendere visibile a tutti quanto la raccolta organica di dati, la loro strutturazione e l'impiego di strumenti avanzati di statistica per l'analisi di dati di partite e giocatori fossero importanti anche applicati alle competizioni sportive. Beane, non potendo competere con il potere d'acquisto dei suoi rivali, adottò un approccio analitico volto a massimizzare il rendimento dei giocatori, portando all'interno del suo team un gruppo di statistici e, grazie all'analisi rigorosa di enormi moli di dati, rivoluzionando poi le tecniche di scouting ed addirittura il modo di guardare il baseball di molti addetti ai lavori. L'approccio di Beane con il tempo diede i



**Figura 3.1:** Billy Beane

suoi frutti, portando, nel 2002, gli Oakland Athletics a vincere l'American League West e riscuotendo attenzione in tutto il mondo. La comunità sportiva osservò con attenzione questo modello e nell'arco di pochi anni modelli simili vennero utilizzati in ambito calcistico, cestistico, nell'hockey ed in molti altri sport.

### 3.1.2 Dati e statistica nello sport moderno

Lo sport moderno è ormai sempre più percepito, visto e raccontato in funzione dei numeri e delle statistiche. Tutto è infatti riconducibile a dati, statistiche, valori comparativi e percentuali, cosa che ha portato, nell'ultimo ventennio, il mondo dello sport a produrre moli di dati sempre più ampie, varie e complesse, e alla conseguente necessità di raccoglierle ed analizzarle. I Big Data sono quindi diventati a tutti gli effetti parte integrante dell'attività di Federazioni nazionali, teams e singoli atleti professionisti, con l'obiettivo di, ottenute quantità significative di dati ed informazioni di interesse, analizzarli ed utilizzare poi quanto appreso in più direzioni. Le conoscenze e le informazioni così ricavate, precise e strutturate, e pertanto cariche di un valore aggiunto importante, vengono utilizzate in modo estremamente vario: dal miglioramento delle tecniche di allenamento e delle prestazioni degli atleti all'analisi delle caratteristiche fisiche, tecniche e tattiche degli avversari, dallo studio approfondito con metodo scientifico di ogni aspetto delle varie discipline al loro cambiamento negli anni, connesso all'introduzione di nuovi materiali, impianti e tecnologie, passando per la prevenzione degli infortuni, la correlazione tra dati e risultati e molto altro ancora. Il tutto con lo scopo di indirizzare scelte, strategie, attività e sinergie.

## 3.2 Il tennis

Il tennis, sport derivato dall'antica pallacorda che trova le sue origine moderne nella seconda metà del XIX secolo in Inghilterra, è uno tra gli sport più diffusi e seguiti al mondo. Si tratta di uno sport globale, con tornei svolti in tutto il mondo e milioni di giocatori provenienti da ogni angolo del globo, seppur con una netta maggioranza di europei ed americani.



**Figura 3.2:** Il Centre Court di Wimbledon: il più famoso stadio di tennis del mondo

Il corpo dirigente nel mondo del tennis è l'ITF, International Tennis Federation, fondata nel 1913, a cui aderiscono 203 associazioni tennistiche nazionali. L'ITF è responsabile dei quattro tornei più importanti della disciplina, ovvero quelli del Grande Slam; i tornei professionistici internazionali più importanti dopo quelli del Grande Slam sono invece organizzati dall'Association of Tennis Professionals (ATP) per gli uomini e dalla Women's Tennis Association (WTA) per le donne, mentre i tornei minori dall'ITF.

### 3.2.1 L'ATP e atptour.com

L'Association of Tennis Professionals, nota con la sigla ATP, è l'associazione che riunisce i giocatori professionistici del tennis maschile di tutto il mondo.



**Figura 3.3:** Logo ATP

L'ATP Tour è, invece, il circuito professionistico mondiale di tennis, organizzato proprio dall'ATP. E' composto dai quattro tornei del Grande Slam, dalle ATP Finals, dai

nove tornei della categoria Mastes 1000 e dai tornei delle categorie 500 e 250. Fanno parte dell'ATP Tour anche i circuiti minori come il circuito Challenger ed il circuito ITF.

Il sito ufficiale dell'ATP, e dell'ATP Tour, è [atptour.com](http://atptour.com), all'interno del quale si possono trovare, aggiornati in tempo reale, risultati, classifiche, statistiche, news ed informazioni di ogni tipo riguardanti giocatori, tornei ed, in generale, il movimento del tennis professionistico maschile mondiale. Sul sito infatti sono consultabili tutti i dati relativi all'ATP Tour, dal 1991 ad oggi, dai risultati di tutti i tornei e di tutte le partite disputate alle classifiche e ai rankings settimana per settimana, passando per le numerose statistiche, frutto dei dati raccolti ed analizzati nel tempo, riguardanti le prestazioni dei giocatori anno per anno e superficie per superficie, i colpi come servizio e risposta, ed altri interessanti parametri legati al gioco.

The screenshot shows the ATP Tour website home page. At the top, there is a dark blue navigation bar with the ATP logo and various menu items: SCORES, STATS, RANKINGS, PLAYERS, TOURNAMENTS, NEWS, VIDEO, and BREAK POINT. Below this, there are several promotional banners and sections. On the left, a large banner features Carlos Alcaraz holding a trophy, with the headline 'Alcaraz Wins Buenos Aires Title In Season Debut'. Below this is a 'Mover of the Week' section for Daniil Medvedev. The central part of the page is dominated by the 'CURRENT TOURNAMENT' section for the 'Rio Open presented by Claro', which includes details about the location (Rio de Janeiro, Brazil), dates (February 20 - February 26 2023), and prize money (\$2,178,980). To the right, there is a 'SCORES' section with a table of match results and a 'HOW TO WATCH' section. The bottom of the page features a 'HEADLINES' section with video thumbnails and a 'Prize Money' section.

Figura 3.4: Home page di [atptour.com](http://atptour.com)

Il sito si presenta quindi a tutti gli effetti come un grandissimo contenitore di dati ed informazioni, un database che contiene al proprio interno tutti i dati e le statistiche del tennis professionistico maschile degli ultimi trent'anni. Risulta essere quindi ottimale per quanto andremo a svolgere, compiendo operazioni di Web Scraping proprio da questo sito.



# Capitolo 4

## Sviluppo dello scraper

Viene ora presentato e descritto il progetto sviluppato. Si tratta di uno scraper programmato per poter, grazie ad esso, fare Web Scraping dal sito dell'ATP, [atptour.com](http://atptour.com) precedentemente presentato, andando ad estrarre dati ed informazioni che, una volta salvati, verranno utilizzati come materiale di analisi per poter discutere di alcuni concetti che ci si prefissa di poter verificare proprio grazie alla raccolta ed analisi di questi dati.

### 4.1 Obiettivi

L'obiettivo di questo progetto è poter utilizzare i dati estratti ed analizzati come prove ed argomenti di discussione di alcuni concetti, che ora verranno presentati, legati al servizio, colpo con cui si inizia ogni punto nel tennis.

#### 4.1.1 Il servizio e le diverse superfici

E' concetto diffuso nel mondo del tennis che “il servizio è più efficace se si gioca su superfici veloci”. Il tipo di superficie su cui viene disputato un match, infatti, è una variabile molto importante in quanto le possibili superfici su cui si gioca (cemento outdoor e veloce indoor uniti nella categoria hard, erba e terra battuta) sono molto diverse tra di loro. Ciò che cambia sono la velocità e l'altezza del rimbalzo della pallina sulla superficie di gioco. I campi in erba sono considerati quelli più veloci: la pallina schizza via molto velocemente ed il rimbalzo rimane spesso molto basso. Caratteristiche simili sono quelle dei campi in cemento e dei campi in manti sintetici (utilizzati nei tornei indoor): anche qui infatti il rimbalzo della pallina è abbastanza veloce e non molto alto. Molto diverse sono invece le caratteristiche dei campi in terra battuta: superficie molto lenta dove infatti la velocità della pallina è più ridotta ed il rimbalzo più alto.

Chiaramente la velocità e l'altezza del rimbalzo della pallina non dipendono solo dal tipo di superficie su cui si gioca ma anche e soprattutto da come viene colpita la pallina. Il servizio, il colpo più veloce per eccellenza, è infatti molto importante; avere un servizio potente e preciso agevola molto l'atleta nella conquista del punto in campi veloci come

quelli in erba e quelli hard, dove la pallina schizza via molto veloce e molto bassa. Obiettivo di questo progetto è quindi verificare se quanto descritto fino ad ora è riscontrabile anche nei dati e nelle statistiche che troviamo sul sito dell'ATP.

### 4.1.2 L'evoluzione del servizio nel corso degli anni

Altro concetto molto diffuso che interessa il servizio è il considerarlo “il colpo che, negli ultimi decenni, ha subito l'evoluzione più importante”. Il servizio è il colpo più complicato da eseguire e la sua esecuzione perfetta è molto complessa. E' opinione comune presso gli addetti ai lavori come, in termini di velocità e, soprattutto, efficacia, tale colpo sia migliorato negli anni; considerazione che trova le sue ragioni nell'acquisizione di una tecnica più efficace, nella migliore preparazione fisica dei tennisti moderni e nel progresso dei materiali utilizzati. Altro obiettivo di questo progetto è quindi osservare se l'evoluzione del servizio fin qui descritta, trova conferma nei dati presenti sul sito.

## 4.2 Approccio iniziale

E' importante, prima di passare alla stesura del codice e alla realizzazione del programma, presentare il modo in cui ci si avvicina ad essa, citando quali dati effettivi si intende estrarre e come si intende procedere, specificando tool e librerie che si andranno ad utilizzare.

### 4.2.1 Dati da estrarre






Gli obiettivi descritti in precedenza, per essere raggiunti, necessitano dell'estrazione e del salvataggio (prima di poter passare alla fase di analisi) delle stesse tipologie di dati. I dati in oggetto sono i dati e le statistiche relative al servizio che troviamo nel sito dell'ATP, [atptour.com](http://atptour.com), alla sezione “Stats - Leaderboards - Serve Leaders”. In particolare, i dati che si andranno ad estrarre sono quelli del “Serve Rating” e dell’“Avg. Aces/Match”, anno per anno e superficie per superficie.

Notiamo infatti come, in questa sezione del sito, i dati siano organizzati in tabelle che cambiano a seconda della combinazione dei valori scelti nei menù a tendina ad inizio pagina; di nostro interesse sono gli ultimi due menù a discesa, dai quali è possibile selezionare l'anno (dal 1991 ad oggi) e la superficie (tutte, terra battuta, erba, hard) di cui si vogliono conoscere i dati. La combinazione di questi valori (più un terzo riguardante la classifica dell'avversario contro cui sono stati ottenuti tali dati, che però per noi sarà fisso sull'opzione “contro tutti i giocatori”) origina delle tabelle, come si può vedere in Figura 4.1, con presenti i dati dei giocatori con il miglior servizio in base ad anno e superficie scelti, ordinati per “Serve Rating” decrescente (opzione comunque modificabile).

Il “Serve Rating” ed l’“Avg. Aces/Match” non sono però gli unici dati che compaiono in queste tabelle; oltre ad essi e alle generalità dell'atleta compaiono infatti altri dati e

statistiche: “% 1st Serve”, “% 1st Serve Points Won”, “% 2nd Serve Points Won”, “% Service Games Won” e “Avg. Double Faults/Match”. Si è scelto di estrarre soltanto il “Serve Rating” e l’ “Avg. Aces/Match” in quanto sono i due dati più significativi.

The screenshot shows the ATP Stats website interface. At the top, there are navigation links: Landing, Leaderboards, Serve Tracker, Performance Zone, Win/Loss, Stats #1s. Below these are three tabs: Serve Leaders (selected), Return Leaders, and Under Pressure Leaders. There are three dropdown menus: 'Versus All Pl...', '2022', and 'All Surfaces'. The main heading is 'Versus All Players On All Surfaces For 2022'. The table below has the following columns: Serve Standing, Player, Serve Rating, % 1st Serve, % 1st Serve Points Won, % 2nd Serve Points Won, % Service Games Won, Avg. Aces/Match, and Avg. Double Faults/Match. The 'Serve Rating' column is highlighted in green.

Serve Standing <sup>®</sup>	Player	Serve Rating <sup>®</sup>	% 1st Serve	% 1st Serve Points Won	% 2nd Serve Points Won	% Service Games Won	Avg. Aces/Match	Avg. Double Faults/Match
1	 John Isner	314.0	68.4%	80.7%	53.5%	91.7%	22.4	2.7
2	 Nick Kyrgios	308.1	68.0%	78.8%	55.9%	92.9%	15.9	3.4
3	 Reilly Opelka	307.1	65.2%	79.5%	57.8%	90.8%	15.9	2.1
4	 Hubert Hurkacz	299.1	63.4%	78.2%	55.3%	90.5%	13.3	1.6
5	 Matteo Berrettini	292.0	64.0%	78.3%	51.6%	87.8%	12.4	2.1

**Figura 4.1:** Parte della tabella con i dati del servizio relativi al 2022 e a tutte le superfici

L’ “Avg. Aces/Match” è infatti il dato che per eccellenza indica l’efficacia del servizio, riportando il numero medio di aces a partita (si parla di aces quando un giocatore fa punto direttamente con il servizio, ovvero quando il servizio è talmente forte e preciso che l’avversario non riesce nemmeno a toccare la pallina). Il “Serve Rating”, invece, è un dato che racchiude al suo interno tutti gli altri citati in precedenza (dati che, se presi da soli, non indicherebbero a pieno l’efficacia del servizio), sommando le percentuali sopra elencate ed il numero medio di aces a partita e sottraendo il numero medio di doppi falli per match, fornendo quindi un ottimo indicatore dell’efficienza del servizio.

## 4.2.2 Struttura del sito

Per procedere con la scelta delle librerie e dei tool da usare è bene però prima capire come sono organizzati i dati di nostro interesse all’interno del sito, così da poter scegliere lo strumento più adatto alle nostre esigenze.

Una volta ispezionato il codice HTML della sezione di nostro interesse del sito si può facilmente notare come le tabelle di nostro interesse vengano popolate attraverso uno script JavaScript, vedasi Figura 4.2.

Osserviamo poi com’è strutturata la tabella. Come si può vedere in Figura 4.3 la struttura è molto chiara: gli elementi “tr”, il cui nome della classe è “stats-listing-row”, corrispondono alle varie righe della tabella e, al loro interno, troviamo tutti i dati e le

```

▼ <script nonce="2396acd6-72dc-4149-9cd5-01e52e761ebd">
  var headerLabels = {
    player : 'Player',
    serveRating : 'Serve Rating<sup>&copy;</sup>',
    firstServePct : ' % 1st Serve',
    firstServePointsWonPct : '% 1st Serve Points Won',
    secondServePointsWonPct : '% 2nd Serve Points Won',
    avgAcesPerMatch : 'Avg. Aces/ Match',
    avgDblFaultsPerMatch : 'Avg. Double Faults/Match',
    serviceGamesWonPct : '% Service Games Won',
  }

```

**Figura 4.2:** Parte dello script che genera le tabelle

statistiche, tra cui le generalità del tennista ed i valori che ci interessano, il “Serve Rating” e l’“Avg. Aces/Match”, rispettivamente il terzo e l’ottavo elemento “td” (i primi due sono il numero di riga nella tabella ed il nome del giocatore).

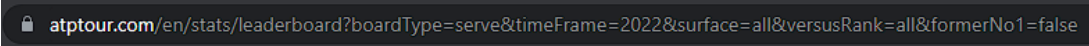
```

▼ <tbody id="leaderboardTable">
  ▼ <tr class="stats-listing-row">
    <td>1</td>
    ▼ <td class="leaderboard-info">
      ▼ <div class="player-leaderboard-info">
        
        
        <a class="stats-player-name" href="/en/players/john-isner/i186/overview">John Isner</a>
      </div>
    </td>
    <td data-type="serveRating" data-value="311.1">311.1</td>
    <td data-type="1stServePct" data-value="68.1%">68.1%</td>
    <td data-type="1stServePointsWonPct" data-value="79.9%">79.9%</td>
    <td data-type="2ndServePointsWonPct" data-value="53.6%">53.6%</td>
    <td data-type="serviceGamesWonPct" data-value="90.6%">90.6%</td>
    <td data-type="avgAcesPerMatch" data-value="21.5">21.5</td>
    <td data-type="avgDblFaultsPerMatch" data-value="2.6">2.6</td>
  </tr>
  ▶ <tr class="stats-listing-row">...</tr>
  ▶ <tr class="stats-listing-row">...</tr>
  ▶ <tr class="stats-listing-row">...</tr>
  ▶ <tr class="stats-listing-row">...</tr>
  ▶ <tr class="stats-listing-row">...</tr>
  ▶ <tr class="stats-listing-row">...</tr>

```

**Figura 4.3:** Codice HTML del body delle tabelle

Altra cosa a cui prestare attenzione è come cambia il link, vedi Figura 4.4, rispetto alla tabella chi si sta visualizzando. E’ facile vedere come ogni tabella “generi” un link diverso dalle altre, ed è altrettanto semplice vedere come questo link abbia una struttura fissa, dove le uniche parti che cambiano sono il numero che segue “timeFrame=”, relativo all’anno selezionato nel rispettivo menù a tendina, e la stringa che segue “surface=”, ad indicare la superficie selezionata (“all” per tutte le superfici, “clay” per la terra battuta, “grass” per l’erba e “hard” per le superfici veloci).



**Figura 4.4:** Link alla tabella con i dati del servizio relativi al 2022 e a tutte le superfici

### 4.2.3 Librerie e tool scelti

Apprese la tipologia dei dati da estrarre e la struttura del sito, ed in particolare delle sezioni nelle quali sono presenti i dati di nostro interesse, è bene ora procedere con la scelta delle librerie e dei tool più adatti alle nostre esigenze.

Dovendo estrarre dati presenti in tabelle popolate attraverso uno script JavaScript ed avendo a che fare con pagine web dinamiche, risulta immediata la scelta di utilizzare Selenium. Come descritto nel Capitolo 2 infatti, Selenium garantisce una gestione molto più semplice delle pagine web dinamiche ed effettua un vero e proprio rendering della pagina, eseguendo il codice JavaScript.

Avendo poi a che fare con dati in formato tabulare, il tool che meglio si addice a questo è Pandas. La descrizione fatta in precedenza ci ricorda infatti come questo tool offra strutture dati ed operazioni per manipolare tabelle numeriche; ad esempio, Pandas DataFrame, è una struttura dati bidimensionale che permette di lavorare con righe e colonne.

### 4.2.4 Idea di partenza

L'idea di base è quella di realizzare uno scraper, utilizzando la libreria Selenium ed il tool Pandas, e quindi il linguaggio Python, che visiti una ad una tutte le possibili tabelle presenti nel sito, e precedentemente descritte, salvandone i dati di interesse.

Si andrà quindi di volta in volta a modificare l'url da visitare a seconda dell'anno e della superficie, gestendo il tutto in modo iterativo, salvando i dati di un anno per tutte le varie superfici, prima di passare all'anno seguente e fare la stessa cosa.

Per ogni combinazione di anno e superficie verrà creata e salvata una tabella nei fogli di lavoro excel con i dati estratti (numero di riga della tabella, nome e cognome del giocatore, "Serve Rating" e "Avg. Aces/Match" di tale giocatore); tuttavia i dati di nostro interesse sono soltanto il "Serve Rating" e l'"Avg. Aces/Match", che verranno salvati in apposite variabili per contribuire alla media totale del "Serve Rating" e dell'"Avg. Aces/Match", anno per anno e superficie per superficie. Tali medie verranno poi salvate in due fogli di lavoro excel specifici, "Avg\_Serve\_Rating.xlsx" e "Avg\_Aces.xlsx", ossia il materiale che verrà utilizzato in fase di analisi.

## 4.3 Sviluppo programma

Descriviamo ora nel dettaglio i passaggi principali della stesura del codice dello scraper.

### 4.3.1 Accesso al sito

Come prima cosa dobbiamo rendere il nostro programma in grado di poter accedere ad atptour.com, e, nello specifico, alla sezione di nostro interesse; per cui, dopo aver importato le librerie ed i tool che ci servono, definiamo l'url da raggiungere e, essendo

il nostro un url che cambia a seconda delle risorse da estrarre, le parti fisse di esso in apposite variabili, come mostrato in figura 4.5.

```

1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  import pandas as pd
4
5  url = "https://www.atptour.com/en/stats/leaderboard"
6
7  fix1 = "?boardType=serve&timeFrame="
8  fix2 = "&surface="
9  fix3 = "&versusRank=all&formerNo1=false"

```

**Figura 4.5:** Import ed url

Ora, dopo aver definito il PATH di chromedriver.exe (che va fornito come parametro nel momento in cui si crea un'istanza di webdriver.Chrome), definiamo la struttura iterativa del nostro programma, componendo così allo stesso tempo, ed in modo opportuno, l'url specifico da raggiungere. Come possiamo notare in Figura 4.6, per tutti i 32 anni di cui il sito dell'ATP fornisce i dati, dal 1991 ad oggi, visitiamo una ad una tutte le quattro tabelle legate alle diverse superfici (tutte le sueprfici, clay, grass, hard).

```

10  x1 = 1991
11
12  PATH = "C:\Program Files (x86)\chromedriver.exe"
13
14  tab1 = pd.DataFrame(columns= ['Year', 'Avg. SR', 'Avg. SR CLAY', 'Avg. SR GRASS', 'Avg. SR HARD',
15                               'Avg. SR TOP 10', 'Avg. SR CLAY TOP 10', 'Avg. SR GRASS TOP 10', 'Avg. SR HARD TOP 10'])
16  tab2 = pd.DataFrame(columns= ['Year', 'Avg. Aces', 'Avg. Aces CLAY', 'Avg. Aces GRASS', 'Avg. Aces HARD',
17                               'Avg. Aces TOP 10', 'Avg. Aces CLAY TOP 10', 'Avg. Aces GRASS TOP 10', 'Avg. Aces HARD TOP 10'])
18
19  for years in range(32):
20      x11 = str(x1)
21      count = 1
22
23      for surfaces in range(4):
24
25          if(count == 1):
26              x2 = "all"
27
28              tab = pd.DataFrame(columns= ['Number', 'Player Name', 'Serve Rating', 'Avg. Aces/Match'])
29
30              driver = webdriver.Chrome(PATH)
31              url1 = url + fix1 + x11 + fix2 + x2 + fix3
32              driver.get(url1)

```

**Figura 4.6:** Struttura iterativa del codice, creazione driver e tabelle

Tabella per tabella, oltre a creare l'oggetto Pandas DataFrame "tab" per salvare i dati che andremo ad estrarre, creiamo un oggetto "driver" di tipo WebDriver Chrome, che utilizzeremo per assegnare al browser le azioni da compiere, come ad esempio, la prima, visitare il sito dell'ATP all'url che forniamo come parametro.

Notiamo inoltre come, all'esterno del primo ciclo, abbiamo anche già creato le due tabelle nelle quali salvaremo i dati finali, ovvero le medie anno per anno e superficie per superficie del "Serve Rating" e dell'"Avg. Aces/Match", che salveremo in due versioni: media totale e media dei primi 10 giocatori, per avere poi in fase di analisi una lente d'ingrandimento sui giocatori migliori.

### 4.3.2 Estrazione dei dati

Ora il nostro scraper è quindi in grado di visitare tutte le tabelle relative al servizio nella sezione stats/leaderboard sul sito dell'ATP; dobbiamo ora fare in modo che estragga i dati e le statistiche di nostro interesse.

Per prima cosa cerchiamo tutti gli elementi con CLASS\_NAME = “stats-listing-row”, ovvero, come visto in precedenza, tutte le righe di ogni tabella, salvandole in una lista.

```

28 driver = webdriver.Chrome(PATH)
29 url1 = url + fix1 + x11 + fix2 + x2 + fix3
30 driver.get(url1)
31
32 p = driver.find_elements(By.CLASS_NAME, "stats-listing-row")
33 c1 = 0
34 total_sr = 0
35 total_sr_top10 = 0
36 total_aces = 0
37 total_aces_top10 = 0
38 for m in p:
39     c1 += 1
40     c11 = str(c1)
41
42     number = m.find_element(By.XPATH, "//tr[" + c11 + "]/td[1]")
43     name = m.find_element(By.XPATH, "//tr[" + c11 + "]/td[2]")
44     serve_rating = m.find_element(By.XPATH, "//tr[" + c11 + "]/td[3]")
45     aces = m.find_element(By.XPATH, "//tr[" + c11 + "]/td[8]")
46
47     temp = pd.DataFrame([[number.text, name.text, serve_rating.text, aces.text]],
48                         columns = ['Number', 'Player Name', 'Serve Rating', 'Avg. Aces/Match'])
49     tab = pd.concat([tab, temp], ignore_index = True)

```

Figura 4.7: Ricerca ed estrazione dei dati

Andiamo poi, elemento per elemento di questa lista (ossia riga per riga della tabella, giocatore per giocatore), a trovare e salvare in apposite variabili i dati e i valori che vogliamo estrarre. Utilizziamo “find\_element” e “XPATH” per trovare i valori di nostro interesse grazie al percorso fornito; infatti, come visto in precedenza, ogni riga della tabella è un elemento “tr” (per cui l’i-esimo elemento “tr” corrisponderà all’i-esima riga della tabella), ed i valori di nostro interesse sono tutti elementi “td” (il numero di riga è il primo, il nome e cognome del tennista il secondo, il suo “Serve Rating” il terzo e la sua “Avg. Aces/Match” l’ottavo).

### 4.3.3 Salvataggio dei dati

Il prossimo step è quello di salvare nel modo opportuno i dati estratti. Come accennato in precedenza, e come si può vedere nelle ultime righe di codice in Figura 4.7, vengono salvati in “tab” tutti i valori estratti; questo oggetto quindi ci permetterà di ottenere anno per anno e superficie per superficie, delle tabelle con tutti i dati estratti di tutti giocatori presenti nelle varie tabelle relative al servizio che si trovano sul sito. Queste tabelle vengono poi, grazie a riga 67 in Figura 4.8, salvate come file excel nella stessa cartella dove è salvato il nostro programma, ognuna con il proprio nome a seconda dell’anno e della superficie riguardanti (ad esempio la tabella relativa ai dati del 2012 e della superficie hard, si chiamerà “2012\_hard.xlsx”).

I dati che maggiormente ci interessano sono però soltanto il “Serve Rating” e l’“Avg. Aces/Match”, che dobbiamo manipolare affinché contribuiscano alla media, totale e top 10, anno per anno e superficie per superficie, dei rispettivi valori.

Dobbiamo quindi, innanzitutto, come mostrato nelle prime due righe di codice in Figura 4.8, portare i valori estratti in formato float e sommarli alla somma totale di “Serve Rating” e “Avg. Aces/Match”, somma che poniamo uguale a due variabili dedicate alla somma di questi valori per i top 10 nel caso in cui il giocatore di cui stiamo estraendo e salvando i dati sia uno dei primi dieci.

```

50         total_sr += float(serve_rating.text)
51         total_aces += float(aces.text)
52
53         if(c1 <= 10):
54             total_sr_top10 = total_sr
55             total_aces_top10 = total_aces
56
57         avg_sr_all = total_sr/c1
58         avg_sr_all_top10 = total_sr_top10/10
59         avg_aces_all = total_aces/c1
60         avg_aces_all_top10 = total_aces_top10/10
61
62         avg_sr_all = "%.2f" % avg_sr_all
63         avg_sr_all_top10 = "%.2f" % avg_sr_all_top10
64         avg_aces_all = "%.2f" % avg_aces_all
65         avg_aces_all_top10 = "%.2f" % avg_aces_all_top10
66
67         tab.to_excel('' + x11 + "_" + x2 + '.xlsx', index = False)
68         driver.quit()
69         count += 1

```

**Figura 4.8:** Salvataggio dati

Una volta esaurite le righe della tabella, dividendo i valori totali di “Serve Rating” e “Avg. Aces/Match” ottenuti per il numero di righe (ovvero il numero di giocatori presenti in tale tabella) e quelli dei top 10 per 10, otterremo la media, totale e dei top 10, di “Serve Rating” e “Avg. Aces/Match” per l’anno e la superficie in questione. Arrotondiamo poi i valori ottenuti alla seconda cifra decimale per ottenere dei dati omogenei.

Arrivati a questo punto, vedi riga 68 in Figura 4.8, possiamo chiudere tutte le finestre del browser e terminare la sessione del WebDriver. La riga successiva, riga 69, aumenta il contatore che ci permetterà quindi di passare alla tabella successiva, con stesso anno ma superficie diversa (il codice è analogo a quanto mostrato fino ad ora, cambia soltanto la condizione iniziale, come mostrato in Figura 4.9).

```

71         elif(count == 2): 119         elif(count == 3): 173         elif(count == 4):
72             x2 = "clay"    120             x2 = "grass"    174             x2 = "hard"

```

**Figura 4.9:** Elif per diverse superfici

Un’unica differenza è presente alla fine del quarto ed ultimo elif, quello riguardante le superfici hard. Arrivati a questo punto infatti, dobbiamo salvare i valori di media ottenuti nelle rispettive tabelle, create ad inizio programma (vedesi righe 14-16 in Figura 4.6). L’aumento del valore x1 a riga 232, in Figura 4.10, ci permette poi di passare all’anno successivo e ripetere quanto fatto fino ad ora.

Una volta visitate tutte le tabelle di tutti gli anni dal 1991 al 2022 e di tutte le superfici, ed usciti quindi da tutti i cicli, l’ultimo step è quello di salvare come file excel



(vedi ultime due righe di codice in Figura 4.10) nella stessa cartella dove è salvato il nostro programma, le due tabelle ottenute con rispettivamente le medie, totali e top 10, anno per anno e superficie per superficie, di “Serve Rating” e “Avg. Aces/Match”.

```

217     temp1 = pd.DataFrame([[x1, avg_sr_all, avg_sr_clay, avg_sr_grass, avg_sr_hard,
218                          avg_sr_all_top10, avg_sr_clay_top10, avg_sr_grass_top10, avg_sr_hard_top10]],
219                          columns = ['Year', 'Avg. SR', 'Avg. SR CLAY', 'Avg. SR GRASS', 'Avg. SR HARD',
220                                    'Avg. SR TOP 10', 'Avg. SR CLAY TOP 10', 'Avg. SR GRASS TOP 10', 'Avg. SR HARD TOP 10'])
221     temp2 = pd.DataFrame([[x1, avg_aces_all, avg_aces_clay, avg_aces_grass, avg_aces_hard,
222                          avg_aces_all_top10, avg_aces_clay_top10, avg_aces_grass_top10, avg_aces_hard_top10]],
223                          columns = ['Year', 'Avg. Aces', 'Avg. Aces CLAY', 'Avg. Aces GRASS', 'Avg. Aces HARD',
224                                    'Avg. Aces TOP 10', 'Avg. Aces CLAY TOP 10', 'Avg. Aces GRASS TOP 10', 'Avg. Aces HARD TOP 10'])
225     tab1 = pd.concat([tab1, temp1], ignore_index = True)
226     tab2 = pd.concat([tab2, temp2], ignore_index = True)
227
228     tab.to_excel(' + x11 + " " + x2 + '.xlsx', index = False)
229     driver.quit()
230     count += 1
231
232     x1 += 1
233
234     tab1.to_excel('Avg_Serve_Rating.xlsx', index = False)
235     tab2.to_excel('Avg_Aces.xlsx', index = False)

```

Figura 4.10: Creazione tabelle

### 4.3.4 Output

L’output del nostro programma risulterà quindi essere la creazione di tutte le tabelle anno per anno e superficie per superficie menzionate in precedenza, vedi Figura 4.11, (che per comodità raggruppiamo nella cartella “Tabelle anno per anno e superficie per superficie”) e delle tue tabelle principali “Avg\_Serve\_Rating.xlsx” e “Avg\_Aces.xlsx” nella stessa cartella dove è salvato il nostro programma, come mostrato in Figura 4.12.

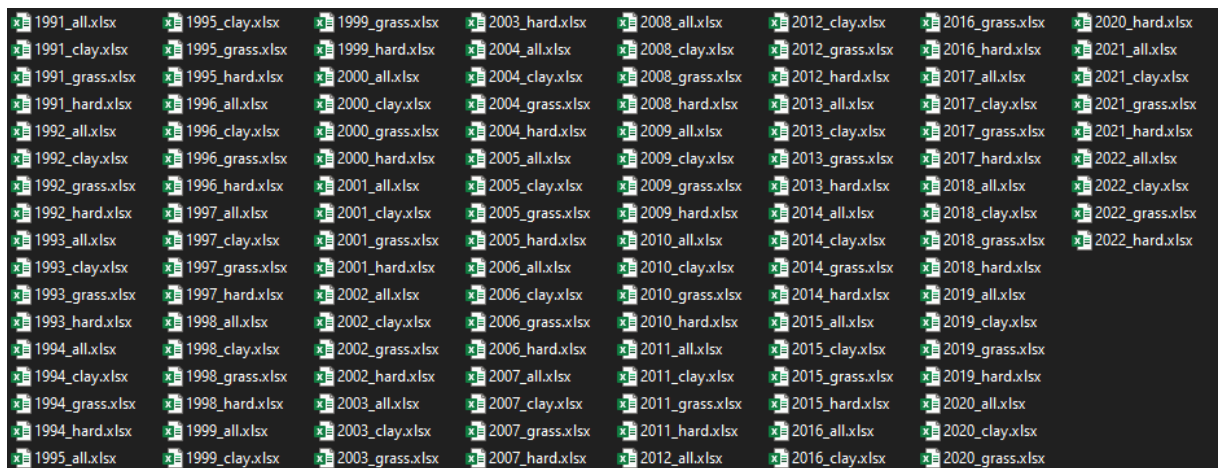


Figura 4.11: Tabelle excel anno per anno e superficie per superficie

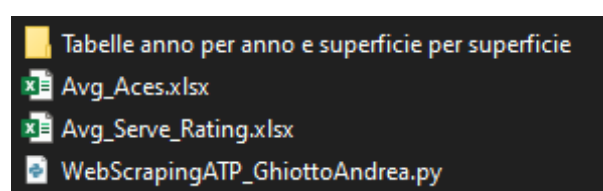


Figura 4.12: Cartella finale con programma Python e tabelle

I file excel delle tabelle anno per anno e superficie per superficie appaiono come mostrato in Figura 4.13, mentre le due tabelle principali “Avg\_Serve\_Rating.xlsx” e “Avg\_Aces.xlsx” appaiono come mostrato in Figura 4.14; tabella dalla struttura analoga, con i propri dati, quella presente in “Avg\_Aces.xlsx”.

Number	Player Name	Serve Rating	Avg. Aces/Match
1	John Isner	319.3	21.4
2	Reilly Opelka	317.8	18.4
3	Nick Kyrgios	304.5	14.6
4	Novak Djokovic	301.6	5.8
5	Hubert Hurkacz	301.0	14.5
6	Alexander Zverev	291.5	8.5
7	Maxime Cressy	290.7	14.5
8	Stefanos Tsitsipas	290.2	7.3
9	Felix Auger-Aliassime	289.2	11.5
10	Matteo Berrettini	289.1	12.5
11	Daniil Medvedev	287.8	10.1
12	Brandon Nakashima	286.4	7.6
13	Taylor Fritz	285.5	10.8
14	Arthur Rinderknech	285.5	9.9
15	Andrey Rublev	284.6	8.4

**Figura 4.13:** Parte della tabella relativa ai dati del 2022 e della superficie hard

Year	Avg. SR	Avg. SR CLAY	Avg. SR GRASS	Avg. SR HARD	Avg. SR TOP 10	Avg. SR CLAY TOP 10	Avg. SR GRASS TOP 10	Avg. SR HARD TOP 10
1991	256,48	244,45	260,79	263,41	278,47	263,67	293,97	281,71
1992	257,07	249,99	262,84	261,24	281,42	271,68	297,96	277,65
1993	256,52	250,59	260,72	257,52	279,2	272,42	295,21	274,82
1994	257,91	250,26	259,09	262,41	280,26	271,37	295,85	278,61
1995	258,66	249,27	255,13	264,48	281,79	271,59	296,33	279,72
1996	259,67	250,19	258,19	264,83	283,29	269,99	295,65	281,57
1997	259,15	248,76	257,08	266,2	283,67	268,39	300,62	282,45
1998	257,93	250,84	256,3	263,72	283,44	269,69	294,32	284,67
1999	258,38	250,78	260,49	266,07	283,06	269,92	301,64	285,77
2000	260,12	251,83	265,89	265,58	282,9	276,79	302,24	280,4
2001	258,6	249,38	262,59	265,62	284,23	274,62	298,01	283,86
2002	260,12	251,9	258,45	266,25	283,69	275,1	294,21	284,81
2003	260,29	252,1	265,31	268,09	283,89	273,42	302,58	284,4
2004	265,53	253,79	269,4	272,68	290,61	275,16	306,39	291,05
2005	263,55	251,18	270,68	268,08	290,93	274,2	305,5	288,29
2006	262,91	253,62	270,6	264,81	288,22	280,03	303,15	284,99
2007	266,47	254,46	272,9	270,73	292,59	276,06	309,21	293,89
2008	266,42	259,23	271,65	272,18	293,02	284,25	309,02	295,1
2009	264,98	258,58	271,3	268,72	293,59	286,65	317,52	291,62
2010	265,73	259,88	273,58	269,64	289,69	285,03	314,81	289,68
2011	263,57	257,2	270,45	268,64	292,19	283,19	311,88	293,95
2012	265,79	259,38	271,1	266,7	293,09	287,39	311,81	293,26
2013	263,48	258,65	273,41	268,88	289,94	284,59	304,19	293,41
2014	266,46	261,03	276,03	267,22	296,71	288,36	315,88	296,5
2015	268,48	264,64	274,85	271,12	299,92	287,33	318,05	299,67
2016	266,68	261,44	272,91	267,73	295,57	290,33	313,18	294,56
2017	267,18	262,02	273,53	268,14	298,8	288,91	313,07	294,09
2018	267,49	262,22	272,25	268,66	297,96	290,98	314,67	296,25
2019	269,32	259,7	271,66	272,88	298,77	283,53	311,15	300,37
2020	270,85	253,29	274,44	297,11	274,44	278,15		300,17
2021	266,64	257,26	270,97	270,51	293,66	285,29	306,17	295,29
2022	269,61	259,9	271,88	273,72	295,71	289,02	305,09	299,49

**Figura 4.14:** Tabella Serve Rating

Nota a margine: non sono presenti i valori di “Avg. SR GRASS” e “Avg. SR GRASS TOP 10” (e nemmeno “Avg. Aces GRASS” e “Avg. Aces GRASS TOP 10” in “Avg\_Aces.xlsx”) per l’anno 2020 in quanto, a causa della pandemia di COVID-19, nel 2020 non si è disputata l’intera stagione su erba.

E’ possibile vedere il codice ed i dati estratti nella loro interezza su GitHub, nella repository indicata al [14] della bibliografia.

# Capitolo 5

## Analisi dei dati estratti

Passiamo ora ad analizzare i dati ed i valori ottenuti grazie allo scraper presentato e descritto nel Capitolo precedente. Per poter discutere, provare, verificare ed argomentare i concetti espressi all’inizio del Capitolo precedente, sono stati prodotti, all’interno dei file excel “Avg\_Serve\_Rating.xlsx” e “Avg\_Aces.xlsx”, dei grafici a partire dai dati estratti e salvati nelle tabelle, così da poterli utilizzare come strumenti in fase di analisi.

### 5.1 Il servizio e le diverse superfici

Il primo dei due concetti è quello riguardante la maggior efficacia del servizio se si gioca su superfici veloci. Ci aspettiamo quindi, per poter verificare questa affermazione, che i valori medi anno per anno del “Serve Rating” e dell’“Avg. Aces/Match” siano nettamente migliori per le superfici veloci (erba ed hard) rispetto alla terra battuta, considerata la superficie più lenta.

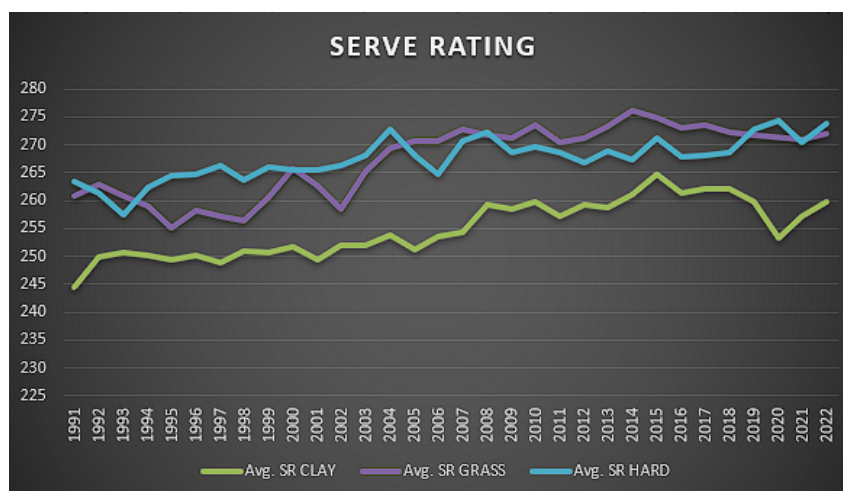
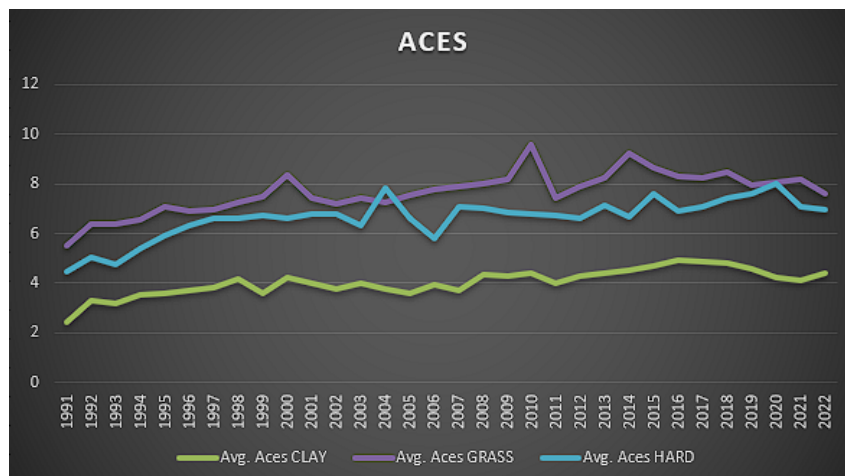


Figura 5.1: Grafico valori medi del Serve Rating nelle diverse superfici

Come possiamo osservare nei grafici in Figura 5.1 ed in Figura 5.2 questa nostra aspettativa è pienamente confermata. Infatti notiamo facilmente come vi sia una netta

separazione tra le linee azzurra e viola (rispettivamente rappresentanti l'erba e le superfici hard) e la linea verde (che rappresenta la terra battuta).



**Figura 5.2:** Grafico valori medi dell’Avg. Aces/Match nelle diverse superfici

Possiamo inoltre notare come l’efficienza del servizio su erba sia mediamente maggiore, sebbene di poco, rispetto alle superfici hard, ponendosi quindi come conferma di quanto descritto in precedenza. Tuttavia questa considerazione non si può considerare assoluta, in quanto vale certamente per l’“Avg. Aces/Match”, vedesi Figura 5.2, ma non del tutto per il “Serve Rating”, vedesi Figura 5.1, permettendoci di affermare come sia maggiore il numero di aces sull’erba rispetto alle superfici hard, ma come invece statistiche come la “% 1st Serve Points Won” e la “% Service Games Won”, valori che contribuiscono a comporre il “Serve Rating”, siano simili in questi due tipi di superficie.

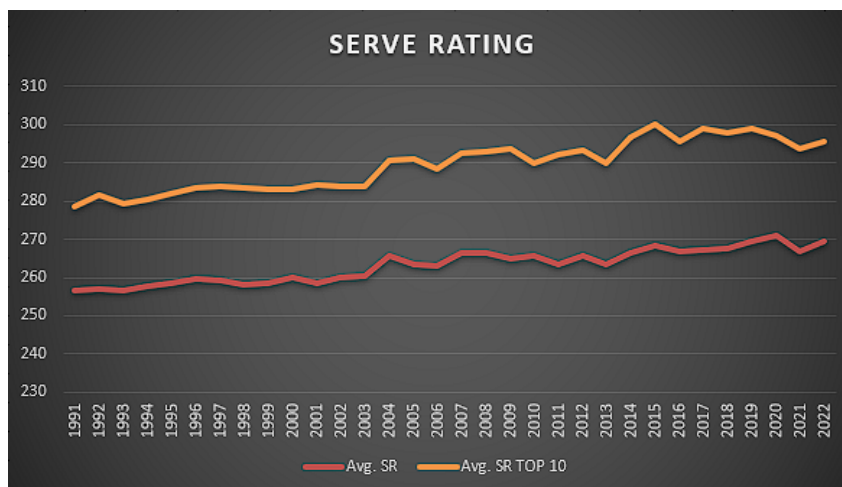
Quest’ultima considerazione si pone quindi come ulteriore prova del fatto che l’efficienza del servizio sia maggiore se si gioca su superfici veloci come erba ed hard, rispetto alla terra battuta.

Non ne riportiamo i grafici, ma, presi i valori medi di “Serve Rating” e “Avg. Aces/Match” dei top 10 superficie per superficie anzichè quelli totali (una sorta quindi di lente d’ingrandimento sui giocatori con i dati migliori nelle varie tabelle), le considerazioni fatte sinora restano tali, con la differenza tra l’efficienza del servizio sulle superfici veloci rispetto alla terra battuta maggiore di quanto visto in precedenza e con l’erba stabilmente in vantaggio sull’hard.

## 5.2 L’evoluzione del servizio nel corso degli anni

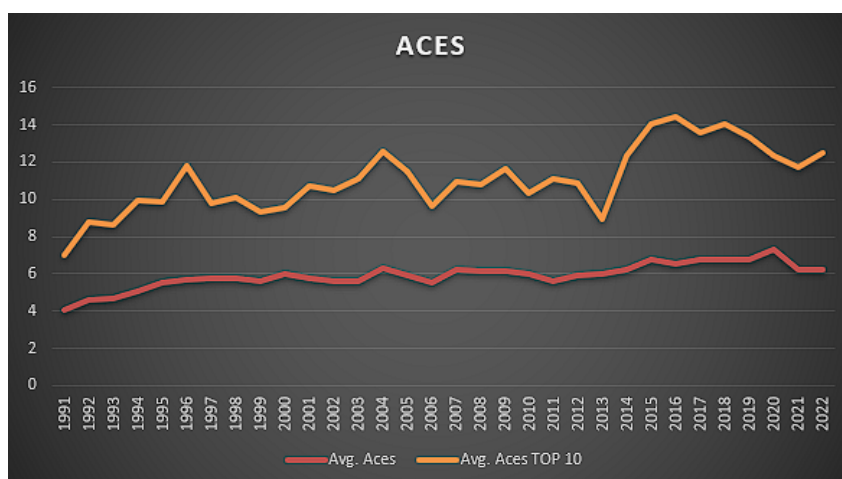
Altro concetto che vogliamo discutere è l’evoluzione del servizio nel corso degli anni. Come descritto in precedenza, infatti, il servizio è considerato il colpo che, negli ultimi decenni, ha subito l’evoluzione maggiore; ci aspettiamo quindi, per poter provare tale affermazione, che i valori medi di “Serve Rating” e “Avg. Aces/Match” siano in costante aumento con il passare degli anni, indipendentemente dalla superficie.

Oltre ai grafici precedenti (Figura 5.1 e Figura 5.2), come strumenti di analisi utilizziamo due ulteriori grafici (Figura 5.3 e Figura 5.4), nei quali sono rappresentati gli andamenti dei valori medi totali rispetto a tutte le superfici di, rispettivamente, “Serve Rating” e “Avg. Aces/Match” nel corso degli anni.



**Figura 5.3:** Grafico andamento Serve Rating negli anni

Possiamo facilmente notare come l'evoluzione del servizio, rappresentata da un significativo aumento della sua efficacia, sia stata graduale e costante nel corso degli anni, e di come questa tendenza sia chiaramente accentuata nelle medie relative ai top 10, rappresentando esse i giocatori con i migliori dati di “Serve Rating” e “Avg. Aces/Match”.



**Figura 5.4:** Grafico andamento Avg. Aces/Match negli anni

Le considerazioni fatte sin qui sono state fatte sulla base della media di “Serve Rating” e “Avg. Aces/Match” rispetto a tutte le superfici. Se però facciamo ora riferimento ai grafici visti in precedenza, ovvero alle Figure 5.1 e 5.2, possiamo notare come l'aumento dell'efficienza del servizio sia aspetto riguardante tutte le superfici, sia quelle che veloci che quelle lente come la terra battuta.

Queste evidenze confermano e provano ulteriormente l'evoluzione del servizio nel corso degli anni e la crescente importanza assunta da questo colpo nel gioco del tennis.



# Capitolo 6

## Conclusioni

Nei Capitoli precedenti si è affrontato il Web Scraping in tutte le sue sfumature, dalle sue origini ad una sua descrizione teorica, dagli strumenti, tool e librerie adatti allo sviluppo di un vero e proprio scraper in grado di estrarre autonomamente ed in modo automatizzato dati dal web, salvandoli in modo opportuno. E' stato inoltre presentato e descritto l'ambito nel quale si è operato, concludendo con un'analisi dei dati ottenuti.

Le principali difficoltà incontrate sono state dovute dall'aver a che fare con pagine web dinamiche e, nello specifico, dal dover estrarre dati presenti all'interno di tabelle popolate attraverso degli script. Ciò infatti, come mostrato in questo elaborato, necessita dell'utilizzo di Selenium, dato che altri tool e librerie più immediati e semplici da usare come BeautifulSoup non riescono a soddisfare tali esigenze.

La ricerca ed estrazione di dati descritta e presentata in questo elaborato si è concentrata su una parte ristretta del sito dell'ATP, ossia quella dei dati statistici legati al servizio; sono però innumerevoli i settori in cui poter addentrarsi con progetti simili di estrazione, salvataggio ed analisi di dati utilizzando gli stessi principi e lo stesso approccio presentati in questo progetto di tesi.

In conclusione possiamo affermare, forti dell'esperienza maturata nello svolgimento di questo progetto e delle nozioni apprese, come il Web Scraping sia uno strumento estremamente utile e di fondamentale importanza, non solo nell'ambito presentato in questo elaborato ma anche in molti altri. Infatti, in un mondo in cui il web è la fonte principale dalla quale ricavare informazioni ed Internet la più grande collezione di dati esistente, avere delle tecniche che permettono di rendere più facilmente fruibile e meno dispendiosa l'estrazione di dati ed informazioni dai siti web è cosa sempre più necessaria e richiesta, grazie alla quale si possono risparmiare tempo e risorse importanti, con il raggiungimento di obiettivi qualificati da mettere a disposizione di chi si trova, nei più svariati settori, nella posizione di fare valutazioni, definire strategie e prendere decisioni.





# Bibliografia

- [1] Scraping Robot (2022), *Web Scraping History: The Origins of Web Scraping*, <https://scrapingrobot.com/blog/web-scraping-history/>
- [2] Vito Lavecchia, *Caratteristiche e Differenza tra Web Crawling e Web Scraping*, <https://vitolavecchia.altervista.org/caratteristiche-e-differenza-tra-web-crawling-e-web-scraping/>
- [3] Ilaria della Queva (2022), *Crawler e Scraper, cosa sono e come funzionano*, <https://cyberment.it/website-security/crawler-e-scraper-cosa-sono-e-come-funzionano/>
- [4] V. Krotov, L. Johnson, L. Silva (2020), *Tutorial: Legality and Ethics of Web Scraping*,
- [5] Requests Documentation, <https://requests.readthedocs.io/en/latest/>
- [6] BeautifulSoup Documentation, <https://beautiful-soup-4.readthedocs.io/en/latest/>
- [7] Selenium Documentation, <https://www.selenium.dev/selenium/docs/api/py/api.html>
- [8] Pandas Documentation, <https://pandas.pydata.org/docs/>
- [9] Numpy Documentation, <https://numpy.org/doc/>
- [10] Matplotlib Documentation, <https://matplotlib.org/stable/index.html>
- [11] Seaborn Documentation, <https://seaborn.pydata.org/>
- [12] William Pearse (2020), *La statistica nello sport*, <https://inomics.com/it/blog/la-statistica-nello-sport-1291601>
- [13] Sito dell'ATP, <https://www.atptour.com/en/>
- [14] Respository GitHub con codice e dati estratti, <https://github.com/AndreaGhi8/WebScrapingATP>