



RELAZIONE TECNICO SCIENTIFICA DELL'UNIVERSO DI ONE PIECE

Giovanni Cornacchia 830631

Simone Farallo 889717

Michele Salvaterra 891109

Obbiettivo

L'obiettivo è creare un database a grafo che contenga informazioni riguardanti tutto l'universo di *One Piece*, comprendendo personaggi e loro dettagli, rating e numero episodi, rating e numero capitoli, saghe, isole, città, razze, ciurme, taglie, prefigurazioni, frutti del mare, nozioni temporali, così da poter costruire un sistema di raccomandazione *content-based*.

L'utente finale di questo sistema si configura come colui che è interessato a visionare contenuti rilevanti dell'anime più acclamato e seguito della storia, potendo analizzare i vari personaggi, creare una serie storica su di essi, visualizzare le valutazioni di ogni episodio, esplorare i vari frutti del mare, le loro tipologie e quant'altro.

Introduzione

One Piece è un manga scritto e disegnato da Eiichiro Oda, serializzato sulla rivista Weekly Shonen Jump dal 22 luglio 1997.

La storia segue le avventure di Monkey D. Luffy, un ragazzo il cui corpo ha assunto le proprietà della gomma dopo aver inavvertitamente ingerito un frutto del diavolo. Raccogliendo attorno a sé una ciurma, Luffy esplora la Rotta Maggiore in cerca del leggendario tesoro 'One Piece' e insegue il sogno di diventare il nuovo Re dei pirati.

- [BeautifulSoup](#)
- [Read HTML](#)
- [Estensione di Chrome “Web Scraper”](#)

1. BeautifulSoup

BeautifulSoup è una libreria di Python finalizzata all'estrazione di dati da file HTML e XML, tra i tre metodi utilizzati è sicuramente il più complesso perché richiede una buona manualità del coding in Python e una conoscenza base del linguaggio HTML per poter estrapolare gli attributi più utili alla ricerca.

È altamente consigliata a tutti coloro che masticano Python in quanto un livello di difficoltà maggiore porta ad una estrazione dei dati più accurata e coerente con il tipo di ricerca che si vuole fare.

2. Read_HTML

Questa funzione ha lo scopo di estrapolare tabelle da una pagina HTML – come si poteva facilmente evincere dal nome – ed è sicuramente il metodo più semplice tra quelli utilizzati come anche il più veloce ma, di contro, forse il più vincolante: vengono estrapolate solo tabelle preimpostate da pagine web, quindi, è richiesto un data cleaning accurato e lascia poco margine di estrapolazione al programmatore.

Altamente consigliata a chi possiede poche basi di programmazione e si accontenta del lavoro già svolto nelle pagine web, avendo così un netto risparmio di tempo.

3. Web Scraper - Free Web Scraping

Free Web Scraper è uno strumento per l'estrazione di dati dal Web con un'interfaccia semplice e intuitiva per il Web moderno, è composto da una semplice interfaccia *point-and-click* e utilizza una struttura modulare composta da selettori che istruiscono lo scraper su come attraversare il sito di destinazione e quali dati estrarre.

L'estrazione dei dati viene eseguita dal browser e non richiede l'installazione di alcunché sul computer, una volta che i dati sono stati estratti, è possibile scaricarli come file CSV o XLSX che possono essere importati in Excel, Google Sheets, ecc.

Essendo che l'estensione Chrome non è reperibile passo per passo negli script Python andiamo a descriverla più approfonditamente.


Questa soluzione può essere interpretata come intermedia tra le due precedentemente elencate, richiede un tempo minimo per l'apprendimento, inoltre, non è comparabile a competenze di programmazione come viene richiesto in *beautiful soup* ma, allo stesso tempo, ha una estrapolazione dati più duttile alle varie casistiche rispetto al `read_html`.



Descrizione dei dataset

Chapter and Episode:

Il seguente dataset riguarda i personaggi canonici e le prime apparizioni. È reperibile a questa pagina: [link](#)



	Name	Chapter	Episode	Year	Note
0	A O	551.0	0460	2009	His name was revealed in the Green data book.
1	Abdullah	704.0	0632	2013	NaN
2	Absalom	444.0	0339	2007	NaN
3	Acilia	706.0	0652	2013	NaN
4	Adele	608.0	0527	2010	Her name was revealed in the Blue Deep data book.
...
443	Zeus	827.0	786.0	2016	His name was revealed in Chapter 844. He was f...
444	Zodia	553.0	462.0	2009	His name was revealed in the Green data book.
445	Zotto	533.0	432.0	2009	His name was revealed in the Vivre Card.
446	Zucca	564.0	489.0	2009	His name was revealed in the Green data book.
447	Zunesha	802.0	751.0	2015	NaN

1317 rows x 5 columns

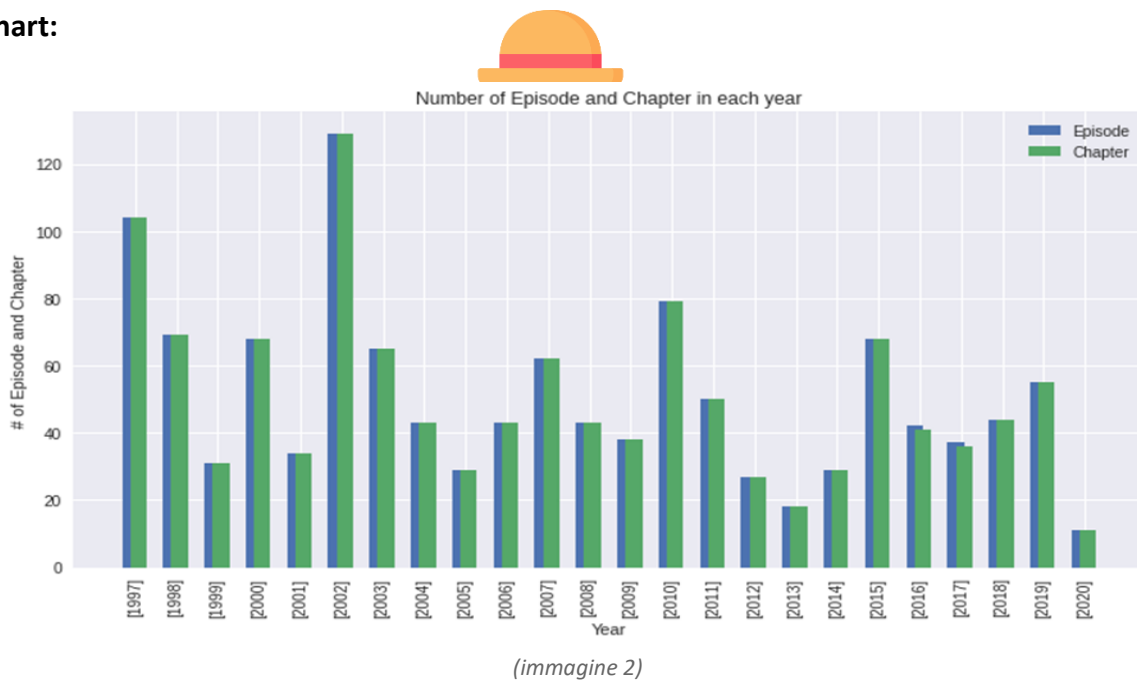
(immagine 1 - 1317 rows x 5 columns)

- *Name*: str, nome del personaggio;
- *Chapter*: int64, numero del capitolo di apparizione;
- *Episode*: int64, numero dell'episodio di apparizione;
- *Year*: int64, anni di apparizione nel capitolo.

Ottenuta tramite la funzione `read_HTML` – Lista dei personaggi di One Piece con l'anno di apparizione nel manga, il volume della loro apparizione e l'episodio nel quale si vedono per la prima volta.

Data Visualization

Barchart:



Come si può notare dal barchart il lavoro svolto dall'autore Oda Eiichirō e l'adattamento anime vanno di pari passo durante gli anni.

Episode Ranking e Rating:

Il seguente dataset riguarda tutti gli episodi dell'anime dall'inizio (1997) fino ad oggi (2022), reperibile in questa pagina: [link](#)



	Rank	Episode	Name_Episode	Start	Total_votes	Average_rating
0	24.129	1	I'm Luffy! The Man Who Will Become the Pirate ...	1999	647.000	7.6
1	29.290	2	The Great Swordsman Appears! Pirate Hunter, Ro...	1999	473.000	7.8
2	32.043	3	Morgan vs. Luffy! Who's This Beautiful Young G...	1999	428.000	7.7
3	28.818	4	Luffy's Past! The Red-haired Shanks Appears!	1999	449.000	8.1
4	37.113	5	Fear, Mysterious Power! Pirate Clown Captain B...	1999	370.000	7.5
...
953	41.448	954	Its Name is Enma! Oden's Meito!	2020	302.000	7.7
954	35.342	955	"A New Alliance?! Kaido's Army Gathers"	2020	407.000	7.4
955	33.715	956	Ticking Down to the Great Battle! The Straw Ha...	2020	353.000	8.2
956	2.940	957	Big News! The Warlords Attack Incident	2021	2.862	9.1
957	14.751	958	"The Legendary Battle! Garp and Roger"	2021	746.000	9.4
958 rows x 6 columns						

(immagine 3 - 958 rows x 6 columns)

Scaricata da kaggle – Lista degli episodi contenente il nome, l'anno, il numero di voti con rispettiva media e il rank che considera le ultime due colonne appena citate e ne dà una valutazione.

- *Rank*: float64, si basa sul numero medio di voti per episodio aggiustato con la valutazione media;
- *Episode*: object, numero dell'episodio;
- *Name_episode*: str, nome dell'episodio;
- *Start*: float64, anno pubblicazione dell'episodio;
- *Total votes*: float64, numero totale di voti;
- *Average_rating*: float64, media del rating per episodio

Data Visualization

Scatterplot:



Si può notare come i primi e gli ultimi episodi siano quelli con più voti (Dimensione) e Rating più elevato, mostrando un forte attaccamento della fan-base verso questa opera.

Ciurme dei pirati:

Il seguente dataset riguarda le varie organizzazioni all'interno dell'universo di One Piece, per esempio le ciurme di pirati, la marina, i nobili mondiali. È reperibile a questa pagina: [link](#)



	Nome_ciurma	Nome
0	Pirati di Cappello di paglia	Monkey D. Rufy
1	Pirati di Cappello di paglia	Roronoa Zoro
2	Pirati di Cappello di paglia	Nami
3	Pirati di Cappello di paglia	Usop
4	Pirati di Cappello di paglia	Vinsmoke Sanji
...
722	Famiglia Kozuki	Kozuki Sukiyaki
723	Famiglia Kozuki	Kozuki Oden
724	Famiglia Kozuki	Kozuki Toki
725	Famiglia Kozuki	Kozuki Momonosuke
726	Famiglia Kozuki	Kozuki Hiyori

676 rows × 2 columns

(Immagine 5 - 676 rows x 2 columns)

Scaricato grazie alla libreria *Beautifulsoup* – Nonostante possa sembrare il dataset più semplice, nella parte di coding viene mostrata la maggiore difficoltà nell'estrarre dati dove il web non facilita l'estrazione e mettendo sotto i riflettori l'elasticità di questa libreria nell'ottenere i nomi dei personaggi con le loro rispettive ciurme.

- *Nome_ciurma*: str, nome della ciurma di appartenenza
- *Nome*: str, nome del personaggio

Dettagli personaggi:

Il seguente dataset riguarda per ogni personaggio la sua data di nascita, l'età e il gruppo sanguigno. È reperibile al seguente [link](#).



	Character	Birth Month	Birth Day	Age	Blood Type
0	Ubau	NaN	NaN	NaN	NaN
1	Noriko	NaN	NaN	NaN	NaN
2	Jigoku Benten	(10) October	18.0	21.0	X (A)
3	Pica	(12) December	14.0	40.0	X (A)
4	Elizabello II	(02) February	2.0	57.0	F (B)

(Immagine 6 - 891 rows x 8 columns)

Ottenuto mediante l'estensione *Web Scraper*, contiene la lista completa di tutti i personaggi all'interno del mondo di One Piece e alcune delle loro caratteristiche principali:

- *Character*: object, nome del personaggio;
- *Bird Month*: object, mese di nascita;
- *Birth Day*: object, giorno di Nascita;
- *Age*: int, età del personaggio;
- *Blood type*: object, gruppo sanguigno;

Isola di nascita:

Il seguente dataset riguarda il mare dove i vari personaggi sono nati, l'isola e la città. È reperibile al seguente [link](#).



	Character	Birth_sea	Birt_Island	Hometown
0	Sanjuan Wolf	West Blue	NaN	NaN
1	T Bone	Grand Line	NaN	NaN
2	Shimotsuki Yasuie	New World	Wano	NaN
3	Gladius	North Blue	NaN	NaN
4	Cavendish	Grand Line	Bourgeois Kingdom	NaN

(Immagine 7 - 377 rows x 6 columns)

Ottenuto mediante l'estensione *Web Scraper*, contiene la lista di tutti i dati riguardanti il mare, l'isola e la città dove i personaggi sono nati.

- *Character*: object, nome del personaggio;
- *Birth_sea*: object, mare di appartenenza;

- *Birt_Island*: object, isola di appartenenza;
- *Hometown*: object, città di appartenenza;

Razza:

Il seguente dataset riguarda la specie e l'altezza. È reperibile al seguente [link](#).



	Character	Race	Rank_Height	Height in Meters	Height in Foot
0	Nico Olvia	Human	311	1.86	6.10
1	Gotti	Human (Cyborg)	103	3.75	12.30
2	Inazuma	Human	208	2.28	7.48
3	Pedro	Mink	203	2.33	7.64
4	Hiking Bear	Animal	48	7.00	22.97

(Immagine 8 - 425 rows × 7 columns)

Ottenuto mediante l'estensione *Web Scraper*, contiene la lista completa della specie e dell'altezza, calcolate in metri e in piedi.

- *Character*: object, nome del personaggio;
- *Race*: object, razza di appartenenza;
- *Rank_Height*: int64, ordine di altezza dal più alto al più basso;
- *Height in Meters*: object, altezza in metri;
- *Height in Foot*: object, altezza in piedi.

Taglie:

Il seguente dataset riguarda l'ammontare delle taglie convertite in euro e dollaro. È reperibile al seguente [link](#).



	Character	Rank_Wanted	Bounty (USD)	Bounty (EUR)
0	Peachbeard	95	\$466,874	€394,049
1	Bepo	144	\$4.49	€3.79
2	Porchemy	141	\$30,526	€25,765
3	Vito	71	\$852,943	€719,897
4	Dorry	66	\$897,835	€757,786

(Immagine 9 165 rows × 6 columns)

Ottenuto mediante l'estensione *Web Scraper*, contiene la lista completa delle taglie e del loro ammontare, sia in dollari che in euro.

- *Character*: object, nome del personaggio;
- *Rank_Wanted*: int64, classifica dei personaggi in base alla taglia più alta;
- *Bounty (USD)*: object, valore taglia in dollari;
- *Bounty (EUR)*: object, valore taglia in euro.

Devil Fruit:

Il seguente dataset riguarda i frutti del mare (tradotti in giapponese come frutti del diavolo), la loro classe di appartenenza e se sono stati risvegliati i poteri. È reperibile al seguente [link](#).



	Character	Devil Fruit	Devil Fruit Class	deceased	Awakening_fruit_of_devil
0	Tamago	Tama Tama no Mi	Zoan	✓	X
1	Chaka	Inu Inu no Mi, Model: Jackal	Zoan	✓	X
2	Donquixote Rosinante	Nagi Nagi no Mi	Paramecia	X	X
3	Blamenco	Poke Poke no Mi	Paramecia	✓	X
4	Epoida	(unknown)	Zoan	✓	X


(Immagine 10 - 183 rows × 8 columns)

Ottenuto mediante l'estensione *Web Scraper*, sono frutti straordinari che donano una grande varietà di poteri, unici per ogni frutto, che si dividono in tre categorie: Paramisha, Zoo Zoo e Rogia.

- *Character*: object, nome del personaggio;
- *Devil Fruit*: object, nome frutto del mare;
- *Devil Fruit Class*: object, classe del frutto del mare;
- *Deceased (EUR)*: object, personaggio vivo o morto;
- *Awakening_Fruit_of_the_Devil*: object, risveglio del frutto del mare.

Tipo di Haki:

Il seguente dataset contiene la lista dei personaggi che hanno sviluppato i vari tipi di Haki. È reperibile al seguente [link](#).



	Character	Armament Haki	Observation Haki	Conquerors Haki
0	Draw	✓	NaN	NaN
1	Vergo	✓	✓	NaN
2	Charlotte Daifuku	✓	✓	NaN
3	Jaguar D. Saul	NaN	NaN	NaN
4	Page One	✓	✓	NaN


(Immagine 11 - 132 rows × 4 columns)

Ottenuto mediante l'estensione *Web Scraper*, Haki nel mondo di One Piece è un potere misterioso che consente all'utente di utilizzare la propria energia spirituale e si divide in 3 categorie.

- *Character*: object, nome del personaggio;
- *Armament Haki*: object, Haki dell'armatura;
- *Observation Haki*: object, Haki dell'osservazione;
- *Conquerors Haki*: object, Haki del Re conquistatore.

Saghe:

Il seguente dataset contiene la lista delle saghe di One Piece, il capitolo ed l'episodio iniziale, il numero totale di capitoli ed episodi, la percentuale di manga e anime completato rispetto all'intera storia. È reperibile al seguente [link](#).



Saga N°	Arco narrativo	Start on Chapter	Total Chapters	Total Pages	Manga %	Start on Episode	Total Episodes	Anime %
2	1.0 Romance Dawn Arc	1	7	178	0.9%	1	3	0.3%
20	2.0 Orange Town Arc	8	14	273	1.4%	4	5	0.5%
23	3.0 Syrup Village Arc	22	20	396	2.0%	9	10	1.0%
26	4.0 Baratie Arc	42	27	514	2.7%	19	12	1.2%
17	5.0 Arlong Park Arc	69	27	514	2.7%	31	15	1.5%

(Immagine 12 - 49 rows × 9 columns)

Ottenuto mediante l'estensione *Web Scraper*

- *Saga N°*: float64, numero saga;
- *Arco narrativo*: object, arco narrativo;
- *Start on Chapter*: int64, capitolo di inizio saga;
- *Total Chapters*: int64, totale capitoli saga;
- *Total Pages*: object, pagine totali all'interno della saga;
- *Manga %*: object, percentuale di manga che questa saga occupa rispetto al totale;
- *Start on Episode*: int64, episodio di inizio saga;
- *Total Episode*: int64, totale episodi saga;
- *Anime %*: object, percentuale di anime che questa saga occupa rispetto al totale.

Foreshadow:

Il seguente dataset contiene la lista dei vari Foreshadow che l'autore del manga ha creato. Questa tabella presenta le più grandi e importanti prefigurazioni di One Piece. Ognuna di esse è presentata con l'ambientazione e il payoff, insieme ai rispettivi capitoli in cui si verificano. Le ultime colonne mostrano il tempo impiegato per il payoff in capitoli, giorni e anni.

Reperibile al seguente [link](#)



	scan manga	Setup	scan manga rivelato	Payoff	Chapters Later	Days Later	Years Later
0	159	Ace gives Luffy a piece of paper	489	Vivre Cards are explained	330	2,667	7.3
1	754	Kanjuro draws badly with his left hand	976	Kanjuro draws well when using his right hand	222	2,079	5.7
2	555	Ace reveals having learned how to make a kasa ...	912	Tama reveals having met Ace in Wano	357	3,248	8.9
3	5	Koushirou uses two crossed swords as a symbol ...	942	A Wano Samurai from 25 years ago appears using...	937	7,931	21.7
4	154	There was no snow on the day Ace visited Drum ...	NaN	Ben Beakman reveals that melted because Ace wa...	NaN	6,454	17.7

(immagine 13 - 61 rows x 7 columns)

Ottenuto mediante l'estensione *Web Scraper*, costituisce un'anticipazione di eventi futuri.

- *Scan manga*: int64, numero capitolo del manga;
- *Setup*: object, prefigurazione evento;
- *Scan manga rivelato*: object, numero capitolo del manga;
- *Payoff*: object, spiegazione dell'evento prefigurato;
- *Chapter Later*: object, somma totale dei capitoli dalla prefigurazione alla spiegazione;
- *Days Later*: object, giorni totali dalla prefigurazione alla spiegazione;
- *Years Later*: float64, anni totali dalla prefigurazione alla spiegazione;

Manga:

Il seguente dataset contiene le informazioni rilevanti sui capitoli del manga usciti, con tanto di titoli, data di pubblicazione e numeri di pagine per capitolo



number_chapter		Title_chapter	release_date_chapter	page_chapter
209	1	Romance Dawn - The Dawn of the Adventure	July 19, 1997	53
531	2	That Guy, "Straw Hat Luffy"	July 28, 1997	23
1025	3	Introducing "Pirate Hunter Zoro"	August 4, 1997	21
39	4	Marine Captain "Axe-Hand Morgan"	August 11, 1997	19
848	5	Pirate King and Master Swordsman	August 25, 1997	19
...
954	1045	Next Level	April 4, 2022	19
349	1046	Raizo	April 11, 2022	17
2	1047	The Sky Over the Capital	April 25, 2022	19
1005	1048	Twenty Years	May 9, 2022	17
878	1049	The World That Should Be	May 16, 2022	17

1049 rows x 4 columns

(Immagine 14 - 1049 rows x 4 columns)

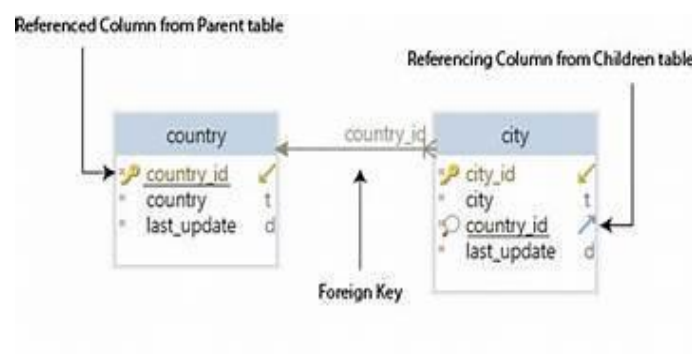
Ottenuto mediante l'estensione *Web Scraper*.

- *Number_Chapter_manga*: int64, numero capitolo del manga;
- *Title_Chapter*: object, titolo capitolo del manga;
- *Release_date_chapter*: object, giorno, mese e anno di uscita;
- *Page_chapter*: int64, totale pagine capitolo;



Processo di cleaning e integrazione

Con Data Cleaning si intende l'insieme di tutti i passaggi necessari per trasformare il dato grezzo in informazione e, quindi, in conoscenza. Con Integrazione, invece, si intende la fase di merging grazie alla quale, prendendo in considerazione il vincolo di integrità referenziale reperibile da due tabelle, è possibile unirle.



(Immagine 15 - integrazione)

Le analisi esplorative dei dataset scaricati sono state svolte utilizzando il linguaggio di programmazione Python. Il codice è visionabile ([link.ipynb](#)).

Prima fonte:

Per quanto riguarda i personaggi, il processo di cleaning e l'integrazione hanno portato all'unione e la pulizia dei seguenti dataset: *"dettagli personaggi"*, *"Isola di nascita"*, *"Pesorazza"*, *"Taglia"*, *"Fruttideldiavolo"*, *"Haki"*.

Dopo vari processi di Data Cleaning abbiamo unito i 6 dataset sopracitati attraverso la funzione merge con il vincolo di integrità referenziale che corrisponde a Character, ottenendo:



	Character	Birth Month	Birth Day	Age	deceased	Birth_sea	Birt_Island	Hometown	Blood Type	Height in Meters
0	Ubau	NaN	NaN	NaN	NaN	Paradise	Skypiea	NaN	NaN	NaN
1	Noriko	NaN	NaN	NaN	✓	NaN	NaN	NaN	NaN	NaN
2	Jigoku Benten	(10) October	18.0	21.0	NaN	NaN	NaN	NaN	X (A)	2.25
3	Pica	(12) December	14.0	40.0	✓	North Blue	NaN	NaN	X (A)	4.70
4	Elizabello II	(02) February	2.0	57.0	NaN	Grand Line	Prodence Kingdom	NaN	F (B)	4.36
...
960	Charlotte Cadenza	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
961	Gion	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
962	Charlotte Raisin	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
963	Tokikake	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
964	Charlotte Counter	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
964 rows x 22 columns										

(immagine 16 - 964 rows x 22 columns)

La seconda fonte:

“Personaggi” (prima apparizione del personaggio nel Manga e nell’Anime), è stata integrata con la prima, per avere maggiori informazioni.

- Eliminazione dei duplicati (9)
- Unione tramite il comando merge, trattenendo quelli a destra (how = 'right'). Questo ha comportato una perdita di dati che tuttavia riguardano i personaggi apparsi nei Film o nei Filler.



	Character	Birth Month	Birth Day	Age	deceased	Birth_sea	Birt_Island	Hometown	Blood Type
0	A O	(01) January	15.0	NaN	NaN	NaN	NaN	NaN	NaN
1	Abdullah	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Absalom	(12) December	30.0	36.0	X	West Blue	NaN	NaN	F (B)
3	Acilia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Adele	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5 rows x 25 columns									

(immagine 17 - 5 rows x 25 columns)


La terza fonte:

Riguarda il [fuzzymatching](#).

Prendendo in considerazione i dataset ‘personaggi’ e ‘ciurme’ andiamo ad utilizzare la funzione di Fuzzy matching, che quindi utilizza un collegamento probabilistico dei record per ottenere le corrispondenze con la colonna ‘best_match_score’, andando poi a scegliere una soglia arbitraria di

-0.004698851733510213 oltre la quale il matching iniziava a dare errori; ovviamente, in questo caso vengono perse informazioni poiché non tutti i personaggi possiedono una ciurma.

Come ultimo passaggio abbiamo mergiato con il dataset ottenuto dalla seconda fonte andando così ad ottenere il dataset finale:



	Character	Birth Month	Birth Day	Age	deceased	Birth_sea	Birt_Island	Hometown	Blood Type	Height in Meters	...	Devil Fruit	Devil Fruit Class
0	A O	(01) January	15.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1	Abdullah	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
2	Absalom	(12) December	30.0	36.0	X	West Blue	NaN	NaN	F (B)	1.95	...	Suke Suke no Mi	Paramecia
3	Acilia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
4	Adele	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
...
1204	Zeus	(06) June	26.0	NaN	NaN	Grand Line	NaN	NaN	NaN	2.32	...	NaN	NaN
1205	Zodia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1206	Zotto	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1207	Zucca	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1208	Zunesha	(12) December	18.0	1000.0	NaN	Grand Line	NaN	NaN	NaN	35,000.00	...	NaN	NaN

1209 rows x 26 columns

(immagine 18 - 1209 rows x 26 columns)



Data Quality

I Dati sono informazione, la loro mole sta aumentando a macchia d'olio nel tempo ma a volte risultano incomprensibili, errati o ridondanti, ed è qui che la Data Quality viene in soccorso per poter trasformare l'informazione dei dati in conoscenza.

Sono state utilizzate due misure: **l'accuratezza e completezza.**

Accuracy

Esprime quanto i dati considerati siano in grado di rappresentare con precisione l'obiettivo che si sta cercando di raggiungere con l'analisi.

Per svolgere l'accuratezza abbiamo preso arbitrariamente un dizionario di riferimento da poter confrontare con la lista dei nomi ottenuti attraverso la fase di merging, il dizionario in questione è stato estrapolato dal dataset 'Chapter and Episode' con la colonna 'Character', attraverso la libreria sklearn: siamo riusciti ad ottenere un accuracy di 1, questo significa che i nomi hanno una corrispondenza del 100% con il dizionario prestabilito, inoltre, mettendo l'opzione `normalize = False` restituisce il numero di esempi ben classificati che corrisponde proprio a 1209 che non a caso coincide con la lunghezza del nostro dizionario.

Completezza

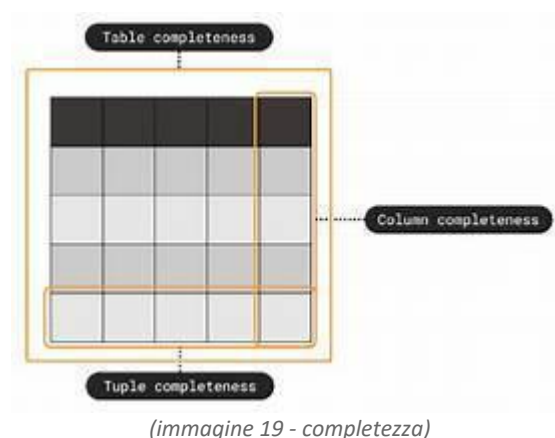
Corrisponde alla copertura con la quale il fenomeno osservato è rappresentato nell'insieme di dati, misura, quindi, la disponibilità di tutti quei dati che ci si aspetta di ottenere da un progetto di studio. Ci sono alcune domande che in genere ci si pone per essere sicuri di avere dati completi: il team ha considerato tutte le caratteristiche di quel particolare mondo che ci aspettiamo di possedere?

Lo sviluppo è stato fatto pensando ad ogni caratteristica che potesse riguardare il mondo di One Piece, per lo meno le caratteristiche fondamentali e di maggiore importanza sono state riportate, essendo un'anime in continua evoluzione da oltre vent'anni, i dati che riguardano le parti non canoniche sono stati omessi.

Questa dimensione è usata per valutare se i dati sono in misura sufficiente per prendere decisioni ed effettuare inferenza.

Come metrica di completezza, è stata valutata la percentuale di osservazioni senza alcun valore mancante sul totale di quelle presenti in ogni dataset calcolando:


- *Tuple completeness*
- *Column completeness*



Tuple completeness

È la percentuale di valori nulli per colonna.

Di seguito è riportata la tabella corrispondente alle colonne, alla somma dei loro valori nulli e la loro percentuale rispetto al totale di righe del dataset.




	Nome_colonna	Valori_nulli	Valori_nulli_%
0	Character	0	0.000000
1	Birth Month	643	0.531844
2	Birth Day	643	0.531844
3	Age	829	0.685691
4	deceased	1031	0.852771
5	Birth_sea	849	0.702233
6	Birt_Island	1025	0.847808
7	Hometown	1098	0.908189
8	Blood Type	835	0.690653
9	Height in Meters	816	0.674938
10	Height in Foot	816	0.674938
11	Rank_Height	816	0.674938
12	Race	830	0.686518
13	Bounty (USD)	1054	0.871795
14	Bounty (EUR)	1054	0.871795
15	Rank_Wanted	1054	0.871795
16	Devil Fruit	1031	0.852771
17	Devil Fruit Class	1031	0.852771
18	Awakening_fruit_of_devil	1031	0.852771
19	Armament Haki	1113	0.920596
20	Observation Haki	1106	0.914806
21	Conquerors Haki	1192	0.985939
22	first_appearance_episode	0	0.000000
23	first_appearance_chapter	2	0.001654
24	first_apparance_year	0	0.000000
25	Nome_ciuma	740	0.612076

(immagine 20 - completezza tuple)

Column completeness:

Percentuale valori nulli per riga.

Nella tabella è possibile vedere la somma di valori nulli per riga e la loro percentuale.



	Nome_riga	valori_nulli_row	valori_percentuale_row_%
0	0	20	0.016543
1	1	21	0.017370
2	2	8	0.006617
3	3	22	0.018197
4	4	22	0.018197
...
1204	1204	14	0.011580
1205	1205	22	0.018197
1206	1206	21	0.017370
1207	1207	22	0.018197
1208	1208	14	0.011580

(immagine 21 - completezza colonne)

Durante lo svolgimento del progetto sono state rispettate delle regole e le convenzioni imposte di neo4j per eseguire query sul dbms (ex: assenza di spazi con sostituzione di ' _')

Dataset estratti e tabelle ponte

Dal dataset `df_personaggi`, dove è stato usato il comando `merge` per unire tutte le informazioni dei personaggi, sono stati ottenuti i seguenti dataset:

- `Sea`: lista di tutti i mari;
- `Class_Fruit`: classi dei frutti del mare;
- `Ciurme`: lista di tutte le ciurme e i gruppi presenti in one piece;
- `Race`: lista contenente tutte le razze presenti.

Successivamente sono state create le tabelle ponte, che verranno utilizzate successivamente nella fase di data modelling, le tabelle create sono:

- `start_on_ep`
- `start_on_chap`
- `rel_foreshadows_scan`
- `rel_foreshadows_scan_revelate`
- `rel_character_ep`
- `rel_character_manga`
- `rel_character_crew`
- `rel_character_sea`
- `rel_character_race`
- `rel_fruit_class`

Data Model

Una volta ottenuti i dati, puliti e opportunamente integrati si è passati alla fase di Data Modeling, in questa fase viene definito il modo in cui si vuole rappresentare l'informazione, è stato quindi deciso di archiviare i dati in un database a grafo, la scelta è stata dettata per la dinamicità e la possibilità di costruire query più o meno complesse e per poter compiere analisi sull'intero database.

I **database a grafo** sono progettati appositamente per l'archiviazione e la navigazione di relazioni. Le relazioni rivestono un ruolo chiave nei database a grafo e buona parte del valore di questi database deriva proprio dalla loro presenza. I database a grafo usano i nodi per archiviare le entità di dati e gli archi per archiviare le relazioni tra le entità. Un arco è definito da un nodo iniziale e da uno finale, dalla tipologia e dalla direzione e può descrivere una relazione genitore/figlio, azioni, la proprietà e il gradimento. Le relazioni che un nodo può avere sono illimitate. In un database a grafo, attraversare i collegamenti o le

relazioni è rapido perché le relazioni tra i nodi non vengono elaborate al momento della query, ma sono già presenti nel database.

Neo4j è un software per basi di dati a grafo open source sviluppato interamente in Java. È un database totalmente transazionale, che viene integrato nelle applicazioni permettendone il funzionamento standalone e memorizza tutti i dati in una cartella. È stato sviluppato dalla Neo Technology, una startup di Malmö, Svezia e della San Francisco Bay Area.

La struttura a grafo di Neo4j si mostra estremamente comoda ed efficiente nel trattare strutture dati come gli alberi o reti distribuite, queste vengono rappresentate con naturalezza da un grafo poiché sono esse stesse dei grafi. L'esplorazione di queste strutture risulta in genere più veloce rispetto a un database a tabelle perché la ricerca di nodi in relazione con un altro nodo è un'operazione primitiva e non richiede più passaggi, in genere tre impliciti in un join di SQL, su tabelle diverse.

Ogni nodo contiene l'indice delle relazioni entranti e uscenti da esso; quindi, la velocità di attraversamento del grafo non risente delle dimensioni complessive ma solo della densità dei nodi attraversati.

Tale modello è quindi costituito da: `

- **Nodi:** descrivono le entità di un dominio, questi possono avere zero o più labels, che definiscono che tipo di nodi sono.
- **Relazioni:** utili a descrivere una connessione tra un nodo di partenza e uno di arrivo, esse devono essere di uno specifico tipo per definire e classificare che tipo di relazioni sono.

Nodi e relazioni: possono avere delle proprietà, coppie di chiave-valore, che li descrivono ulteriormente.

E' stato definito a priori lo schema dei nostri dati anche se in un database a grafo non è necessario.

Nodi e proprietà

In fase di progettazione sono stati creati i seguenti nodi, rappresentiamo in grassetto i label del singolo nodo, Neo4j necessita una specificazione per ogni tipo di proprietà senza la quale, ogni dato viene interpretato come stringa.

- Entità **Characters**: descrive tutte le caratteristiche dei personaggi aggiornate ad aprile 2022
- Entità **Crews**: descrive tutti i gruppi e le ciurme nel mondo di One Piece
- Entità **Episode**: descrive tutti gli episodi dell'anime fino all'episodio 958
- Entità **Foreshadows**: descrive le prefigurazioni all'interno del manga e quando sono queste sono state rivelate
- Entità **Fruit_Class**: descrive tutte le classi che si riferiscono ai frutti del diavolo
- Entità **Manga**: descrive le informazioni riguardante i singoli capitoli del manga
- Entità **Race**: descrive le razze presenti all'interno del mondo di One Piece
- Entità **Saga**: descrive tutte le saghe all'interno sia del Manga che dell'Anime

- Entità **Sea**: descrive tutti i mari presenti nel mondo di One Piece

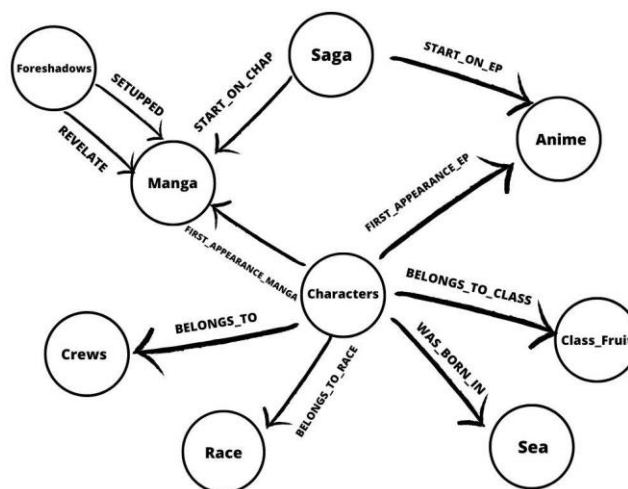
Relazioni

Dopo aver specificato tutte le entità di interesse, si è passati a definire le relazioni fra di esse, quindi sono state definite 10 relazioni:

- Relazione **Saga** –[**START_ON_EP**]->**Episode** : collega ciascuna saga all’episodio dell’anime dove inizia;
- Relazione **Saga**-[**START_ON_CHAP**]->**Manga**: collega ciascuna saga al capitolo del manga dove inizia;
- Relazione **Foreshadows**-[**SETUPPED**]->**Manga**: collega le prefigurazioni al capitolo in cui vengono introdotte;
- Relazione **Foreshadows**-[**REVELATE**]->**Manga**: collega le prefigurazioni al capitolo in cui vengono rivelate;
- Relazione **Characters**-[**FIRST_APPEARANCE_MANGA**]->**Manga**: collega ogni personaggio al capitolo in cui appare per la prima volta;
- Relazione **Characters**-[**FIRST_APPEARANCE_EP**]->**Episode**: collega ogni personaggio all’episodio dell’anime in cui appare per la prima volta;
- Relazione **Characters**-[**WAS_BORN_IN**]->**Sea**: collega ogni personaggio al mare in cui è nato;
- Relazione **Characters**-[**BELONGS_TO_CLASS**]->**Class_Fruit**: collega ogni frutto del diavolo del personaggio alla rispettiva classe del frutto considerato;
- Relazione **Characters**-[**BELONGS_TO**]->**Crews**: collega ogni personaggio alla ciurma a cui appartiene;
- Relazione **Characters**-[**BELONGS_TO_RACE**]->**Race**: collega ogni personaggio alla razza di appartenenza.

Schema del modello

Al termine del processo di Data Modelling, lo schema sarà il seguente:



(immagine 22)

Data Storage

La fase di Data Storage si riferisce alla creazione del DBMS e la memorizzazione dei dati al suo interno.

Le operazioni sono state realizzate tramite Python, attraverso il driver ufficiale Neo4J che permette di stabilire una connessione tra un notebook Python e il DBMS, tutti i dataset sono stati preparati precedentemente nella fase di Pre-Processing.

La fase di modellazione è divisa in tre step:

- 1) **Creazione di vincoli:** Per ogni entità è stato applicato un vincolo di unicità' per la proprietà principale in modo che all'interno del DBMS non è possibile inserire due nodi dello stesso tipo aventi lo stesso valore per la specifica proprietà. Imponendo un vincolo su una proprietà di un nodo viene creato anche un indice riferito a quella proprietà che migliora le performance in fase di interrogazione del DBMS.

esempio di un vincolo:

```
-----  
CREATE CONSTRAINT ON (c:Characters) ASSERT c.name_character IS UNIQUE  
-----
```

- 2) **Creazione dei nodi** è stato effettuato tramite la procedura *LOAD CSV* di Neo4j

Per quanto riguarda i nodi che si riferiscono ai personaggi, dato che non tutti i nodi presentano le stesse proprietà, è stata implementata una procedura che, per ogni record del dataset, ignorasse i valori nulli, in modo che se un nodo avesse delle proprietà con valori nulli, essa non verrebbe creata.

Esempio creazione nodi dei personaggi

```
-----  
LOAD CSV WITH HEADERS FROM 'https://github.com/SimoneFarallo/Data-  
Managment/raw/main/Personaggi_neo4j' AS row
```

```
MERGE (p:Characters {name_character:{row.Character}})
```

```
FOREACH(ignoreMe IN CASE WHEN trim(row.Birth_Month) <> "null" THEN [1] ELSE [] END | SET  
p.birth_month = toString(row.Birth_Month)) .....continua....  
-----
```

- 3) **Creazione delle relazioni** avviene sfruttando le tabelle ponte create in fase di Pre-Processing e integrazione, viene utilizzata quindi la procedura *LOAD CSV* e *MATCH* di neo4j per creare le relazioni.

Esempio relazione saga inizio episodio

```
LOAD CSV WITH HEADERS FROM 'https://github.com/SimoneFarallo/Data-Management/raw/main/rel_saga_ep_neo4j' AS line
```

```
MATCH (s:Saga
{name_saga:toString(line.name_saga)}),(e:Episode{number_ep:toInteger(line.start_on_episode)})
CREATE(s)-[:START_ON_EP]->(e)
```

In totale abbiamo 3401 nodi e 3934 relazioni

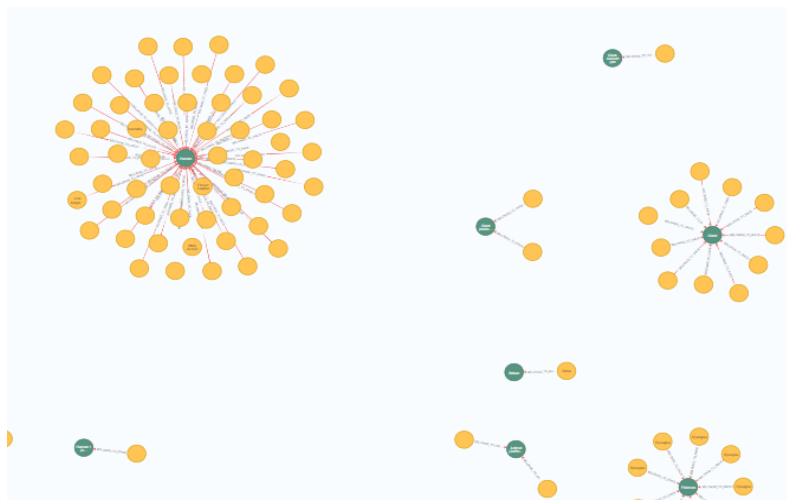
ANALISI ESPLORATIVA

Il linguaggio *Cypher* è usato per interrogare di dati in Neo4j, è ottimizzato per individuare i nodi di interesse e navigare le relazioni tra di essi.

Di seguito mostro alcuni esempi di query:

Quali sono i personaggi più alti di tre metri e a che razza appartengono?

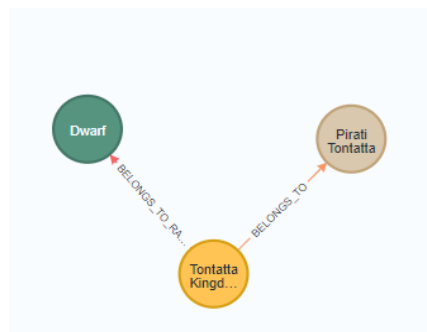
```
MATCH (n:Characters)-[:BELONGS_TO_RACE]->(c:Race)
WHERE n.height_in_meters > 3
RETURN n,c
```



(immagine 23 - output query)

Chi è il personaggio più basso in tutto l'universo one piece e a quale razza e ciurma appartiene?

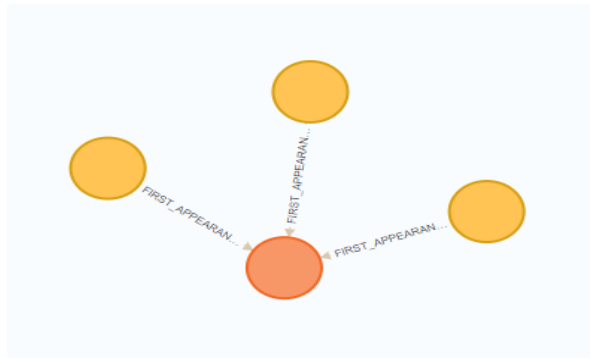
```
MATCH (n:Crews)<-[:BELONGS_TO]- (c:Characters)-[:BELONGS_TO_RACE]->(r:Race)
WITH min(c.height_in_meters) AS min
MATCH (n:Crews)<-[:BELONGS_TO]- (c:Characters)-[:BELONGS_TO_RACE]->(r:Race)
WHERE c.height_in_meters = min
RETURN c,n,r
```



(immagine 24 - output query)

Cerca l'episodio con il rating medio più alto e dimmi che personaggi sono apparsi per la prima volta

```
MATCH (e:Episode)<-[:FIRST_APPEARANCE_EP]-(c:Characters)
WITH max(e.average_rating_ep) AS max
MATCH (e:Episode)<-[:FIRST_APPEARANCE_EP]-(c:Characters)
WHERE e.average_rating_ep=max
RETURN e,c
```



(immagine 25 - output query)

Cerca tutti i personaggi della ‘ciurma delle cento bestie’ che abbiano il frutto del mare e che siano ricercati, ordinandoli in base alla posizione dei più ricercati del mondo e mostrandomi la taglia

MATCH (n:Characters)-[:BELONGS_TO]->(c:Crews{name_crew:'Pirati delle cento bestie'})

WHERE n.devil_fruit is not null and n.rank_wanted is not null

RETURN n.name_character,n.devil_fruit,n.rank_wanted,n.bounty_EUR

ORDER BY n.rank_wanted

n.name_character	n.devil_fruit	n.rank_wanted	n.bounty_EUR
"Queen"	"Ryu Ryu no Mi, Model: Brachiosaurus"	10	"€10,002,778"
"Jack"	"Zou Zou no Mi, Model: Mammoth"	12	"€7,577,862"
"Who's-Who"	"Neko Neko no Mi, Model: Sabertooth"	20	"€4,191,378"
"Black Maria"	"Kumo Kumo no Mi, Model: Rosamygale Grauvogeli"	24	"€3,684,728"
"Sasaki"	"Ryu Ryu no Mi, Model: Triceratops"	26	"€3,623,316"
"Ulti"	"Ryu Ryu no Mi, Model: Pachycephalosaurus"	31	"€3,070,607"

(immagine 26 - output query)

Conclusione e implementazioni future

Dopo la presa visione del report si può concludere che il database prodotto risponde alla domanda di business di descrivere informazioni riguardanti tutto l'universo di One Piece, riuscendo così a costruire un sistema di raccomandazione content-based.

Il gruppo si può ritenere soddisfatto dei risultati ottenuti, in particolare si augura che la fruizione del proprio lavoro possa essere utile a tutti gli appassionati di One Piece che seguono sia l'anime che il manga.

L'utente finale di questo sistema è colui che ha interesse nel visionare contenuti rilevanti di questa opera, potendo analizzare i vari personaggi, creare serie storiche su di essi e visualizzarne le valutazioni di ogni episodio.

I possibili miglioramenti e le implementazioni future sono le seguenti:

- Uniformare il linguaggio all'interno del nostro database completamente in inglese, ottenendo una coerenza di linguaggio e una usabilità più globale del lavoro svolto;
- Aggiornare temporalmente i dati a cadenza annuale;
- Utilizzare strumenti di raccolta dati, diversi dal Web Scraeping, come potrebbero essere le API in modo tale da avere dati più accurati.