

Analisi del mondo anime attraverso AnimeUnity, AnimeWorld e MyAnimeList

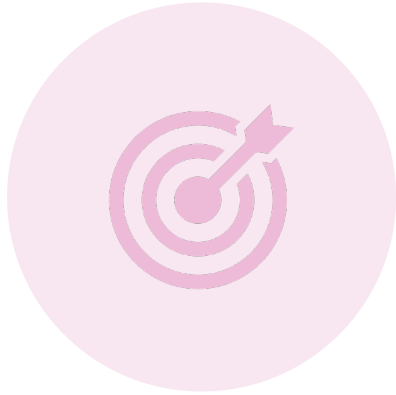
Roberto Ferrari [852220], Samir Doghmi [897358], Stefano Quartuccio
[851599].

Indice

- 01 —> Introduzione
- 02 —> Acquisizione Dati
- 03 —> Database
- 04 —> Data cleaning e Data Integration
- 05 —> Data Quality
- 06 —> Analisi Risultati
- 07 —> Conclusioni e Sviluppi futuri

01 →

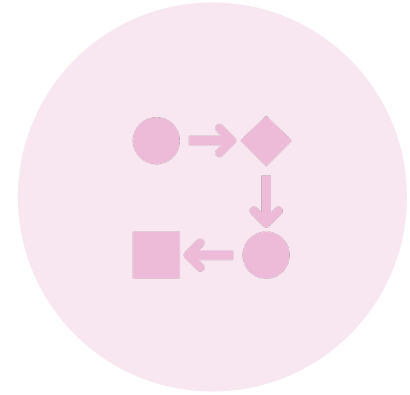
Introduzione



OBIETTIVI



METRICHE DI
VALUTAZIONE



WORKFLOW

Obiettivi

- ❑ La popolarità degli anime può variare a seconda del pubblico e della regione geografica. Il nostro obiettivo è identificare eventuali differenze nella popolarità degli anime tra gli utenti globali e quelli italiani, con particolare focus sulla Top 100 per ciascuna fonte.
- ❑ Le domande centrali a cui vogliamo rispondere sono le seguenti:
 - un titolo amato all'estero è apprezzato anche in Italia?
 - quali sono le caratteristiche che rendono un anime più popolare tra il pubblico?
- ❑ Le fonti di riferimento sono:
 - AnimeUnity (fonte italiana);
 - AnimeWorld (fonte italiana);
 - MyAnimeList (fonte mondiale).

Metriche di Valutazione

- ❑ L'idea di confrontare due piattaforme di streaming con un database online nasce a seguito dell'impossibilità di avere un repertorio italiano che svolga la stessa funzione e sia anche popolare. Le metriche di valutazione utilizzate per confrontare i diversi titoli sono:
 - **Rating**: si riferisce al punteggio numerico che un utente assegna a un anime che ha guardato; la scala di valutazione va da 1 a 10.
 - **Numero di visualizzazioni**: indicano il numero di visite o visualizzazioni della pagina del titolo;
- ❑ Attributi che rispecchiano le metriche, rispettivamente a ciascuna fonte:
 - AnimeUnity: 'Score' (combinazione di 'Favoriti' e 'Visualizzazioni');
 - AnimeWorld: 'Rating' e 'Visualizzazioni';
 - MyAnimeList: 'Score' e 'Popularity'.

Workflow

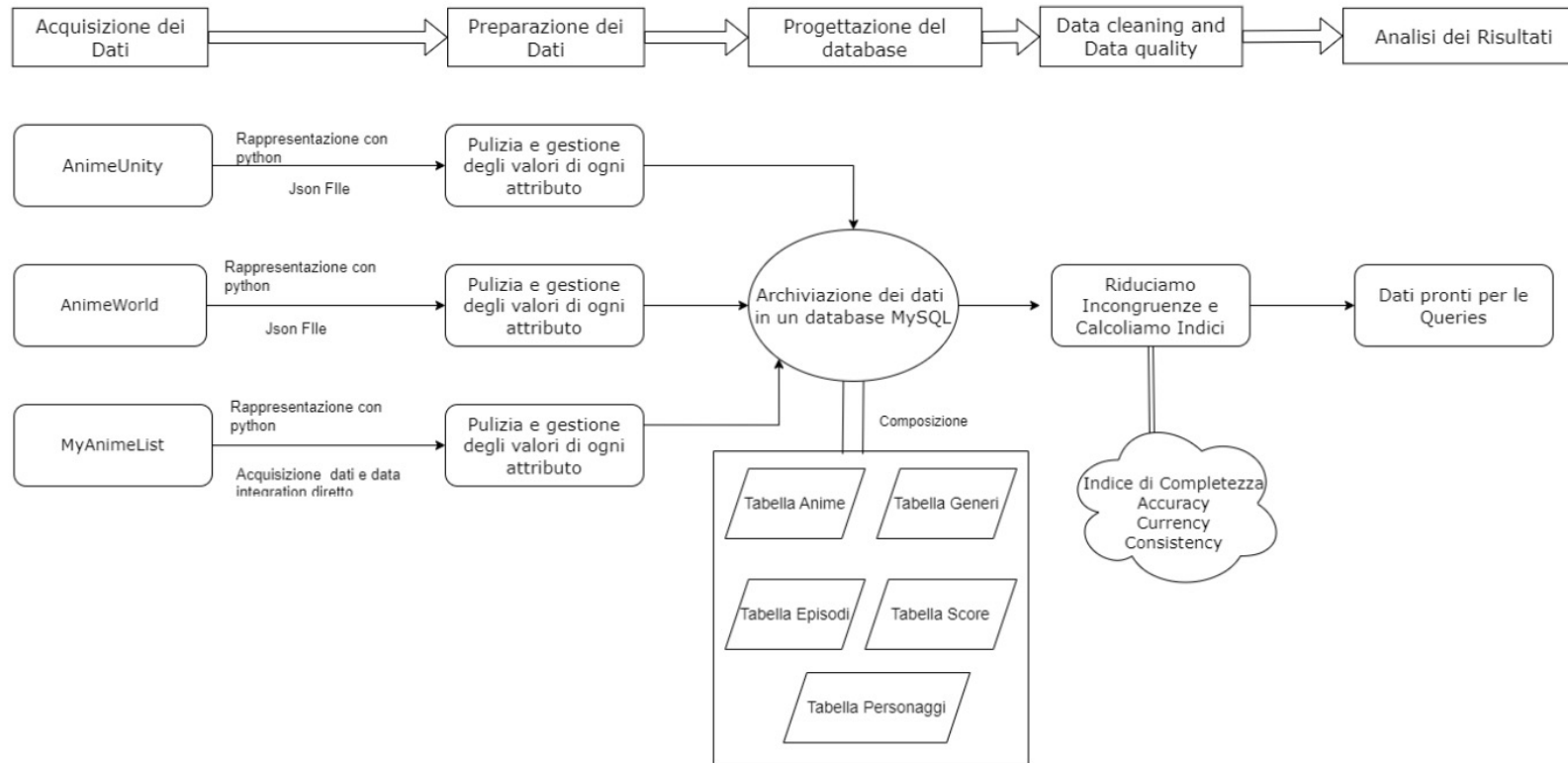


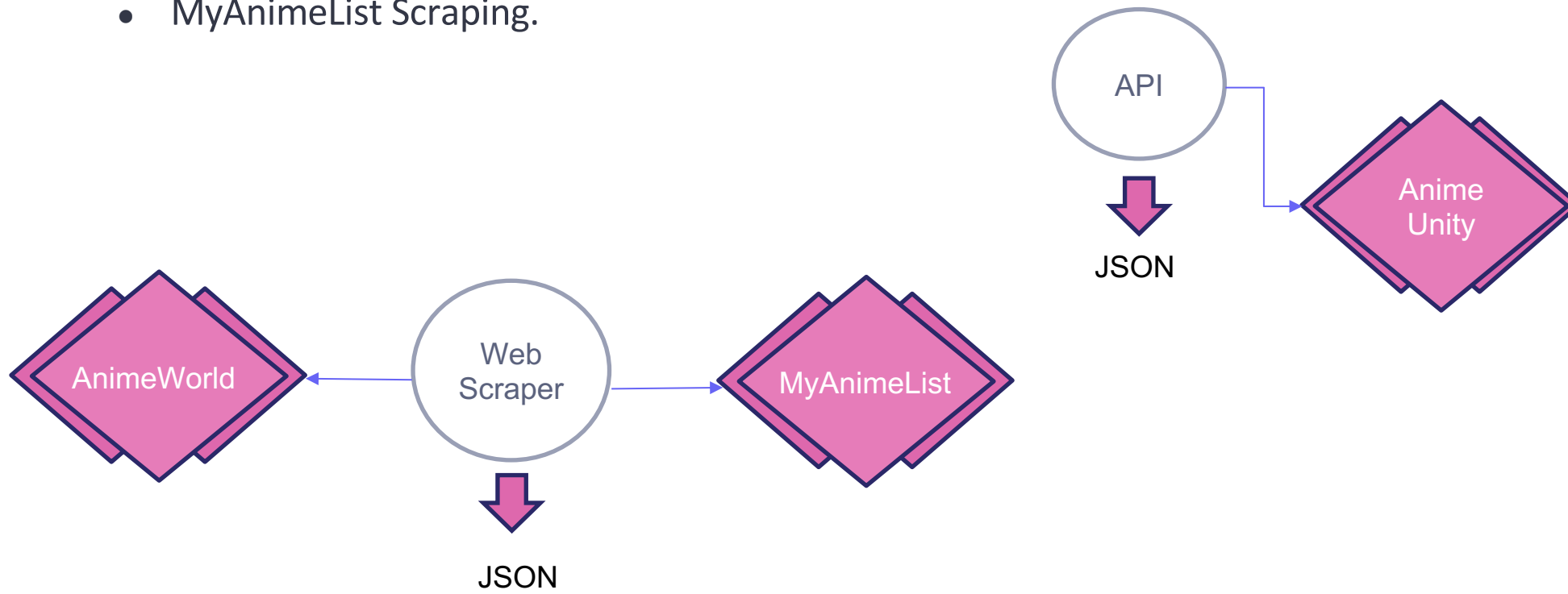
Figure 1: Mappa concettuale fasi progetto

02 →

Acquisizione Dati

Acquisizione Dati

- AnimeUnity API;
- AnimeWorld Scraping;
- MyAnimeList Scraping.



Acquisizione Dati

Esempio di uno Scraper

```
1  # -*- coding: utf-8 -*-
2  from typing import Any, Union
3
4  import scrapy
5
6  class ListAnimeWorld(scrapy.Spider):
7      name = "listAnimeWorld"
8      allowed_domains = ["www.animeworld.tv"]
9      start_urls = [
10         'https://www.animeworld.tv/az-list',
11     ]
12
13     def parse(self, response):
14         for anime_link in response.css(".items .item a ::attr(href)").extract():
15             yield scrapy.Request(response.urljoin(anime_link), callback=self.parse_list_anime_page)
16
17         next_page = response.css("#go-next-page.disabled").extract_first()
18         if not next_page:
19             next_page = response.css("#go-next-page ::attr(href)").extract_first()
20             yield scrapy.Request(response.urljoin(next_page), callback=self.parse)
21
22     def parse_link_url(self, response):
23         p = response.css("#player video source").get()
24         return p
25
26     def parse_list_anime_page(self, response):
27         item = dict[str, Union[list[Any], list[dict[Any, Union[str, Any]]]], Any] = {}
28         array_episodi = []
29         array_generi = []
30
31         for anime in response.css("#animeId .widget-body div.server.active li.episode a "):
32             id_anime = anime.attrib['data-id']
33             episodio = {anime.attrib['data-episode-num']: "https://www.animeworld.tv/api/episode"
34                        + "/" + anime.attrib['data-episode-num'] + "/serverPlayerAnimeWorld?id=" + id_anime}
35             array_episodi.append(episodio)
36
37         for genere in response.css("#main div div.widget.info div div:nth-child(1) div.info.col-md-9 div.row "
38                                   "dl:nth-child(1) dd:nth-child(12) a ::text"):
39             array_generi.append(genere.get())
40
41         item["Data_uscita"] = response.css("#main div div.widget.info div div:nth-child(1) div.info.col-md-9 div.row "
42                                           "dl:nth-child(1) dd:nth-child(6)").get()
43         item["Image_url"] = response.css("#thumbnail-watch img").get()
44         item["MyAnimeList"] = response.css("#mal-button ::attr(href)").extract_first()
45         item["AniList"] = response.css("#anilist-button ::attr(href)").extract_first()
46         item["Stagione"] = response.css("#main div div.widget.info div div:nth-child(1) div.info.col-md-9 div.row "
47                                         "dl:nth-child(1) dd:nth-child(8) a ::text").get()
48         item["Categorie"] = response.css("#main div div.widget.info div div:nth-child(1) div.info.col-md-9 "
49                                         "div.row dl:nth-child(1) dd:nth-child(2)").get()
50         item["Generi"] = array_generi
51         item["Rating"] = response.css("#main div div.widget.info div div:nth-child(1) div.info.col-md-9 div.row "
52                                       "dl:nth-child(2) dd.rating #average-vote ::text").get()
53         item["Visualizzazioni"] = response.css("#main div div.widget.info div div:nth-child(1) div.info.col-md-9 div.row "
54                                                "dl:nth-child(2) dd:nth-child(10) ::text").get()
55         item["Description"] = response.css("#main .desc ::text").extract_first()
56         item["Link-ep"] = array_episodi
57         item["Title"] = response.css("#main div div.widget.info div.head h2 ::text").extract_first()
58         item["Keywords"] = response.css("#tagsReload ::text").extract_first()
59         yield item
60
```

API

```
1  import requests
2  import json
3
4  dataset = {}
5
6  with open('dataAnimeUnity.json', 'r') as infile:
7      dataset = json.load(infile)
8
9  result = []
10 for i in range(0, 3833, 30):
11     url = 'https://www.animeunity.tv/archivio/get-animes'
12     headers = {
13         "authority": "www.animeunity.tv",
14         "Connection": "keep-alive",
15         "Content-Length": "129",
16         "Origin": "https://www.animeunity.tv",
17         "X-Requested-With": "XMLHttpRequest",
18         "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.52 Safari/536.5",
19         "Content-Type": "application/json",
20         "Accept": "*/*",
21         "x-csrf-token": "6USoRVqVlg19qRqkxr2rmlvzuMvRiCXZZF9G70cp",
22         "Referer": "https://www.animeunity.tv/archivio",
23         "Accept-Encoding": "gzip, deflate, sdch",
24         "Accept-Language": "fr-FR, fr;q=0.8, en-US;q=0.6, en;q=0.4",
25         "Accept-Charset": "ISO-8859-1, utf-8;q=0.7, *;q=0.3",
26         "Cookie": "__cf_bm=mbYEdn5X0K580XDgomxtGNGxGVmzHRz.7QBHtfznDg-1670620626-0-AWV770ItQ7E6INxM7V457f9TbK/05Ik37IjNgrY2heRC40dt"
27     }
28     payload = {
29         "offset": str(i)
30     }
31     r = requests.post(url, data=json.dumps(payload), headers=headers)
32
33     resp_json = r.json()
34
35     result.extend(resp_json["records"])
36     print(resp_json["records"])
37
38
39
40 dataset["records"] = result
41
42 jsonFile = open("dataAnimeUnity.json", "w")
43 jsonFile.write(json.dumps(dataset))
44 jsonFile.close()
45
```

03



Database

Scelta del Database



Struttura dei dati e relazioni: Il modello relazionale permette di organizzare i dati in tabelle e stabilire relazioni tra di esse. Poiché ogni anime è considerato come un'entità indipendente e ha diverse caratteristiche associate, come il tipo, lo studio di produzione e il numero di visualizzazioni.



Informazioni chiare sugli attributi: Disponiamo di informazioni chiare sul significato di ciascun attributo e riteniamo che questi dati non cambieranno significativamente nel tempo. Di fatto, la frequenza settimanale di rilascio degli episodi, se si considera che una singola stagione di un anime conta in media 12 episodi.



Integrità dei dati: abbiamo definito dei vincoli di integrità come l'utilizzo di chiavi esterne che garantiscono le relazioni tra le tabelle e le chiavi interne che assicurano l'unicità di ogni riga inserita. In questo modo è possibile fare joints multi-tabella e ottenere risultati complessi.



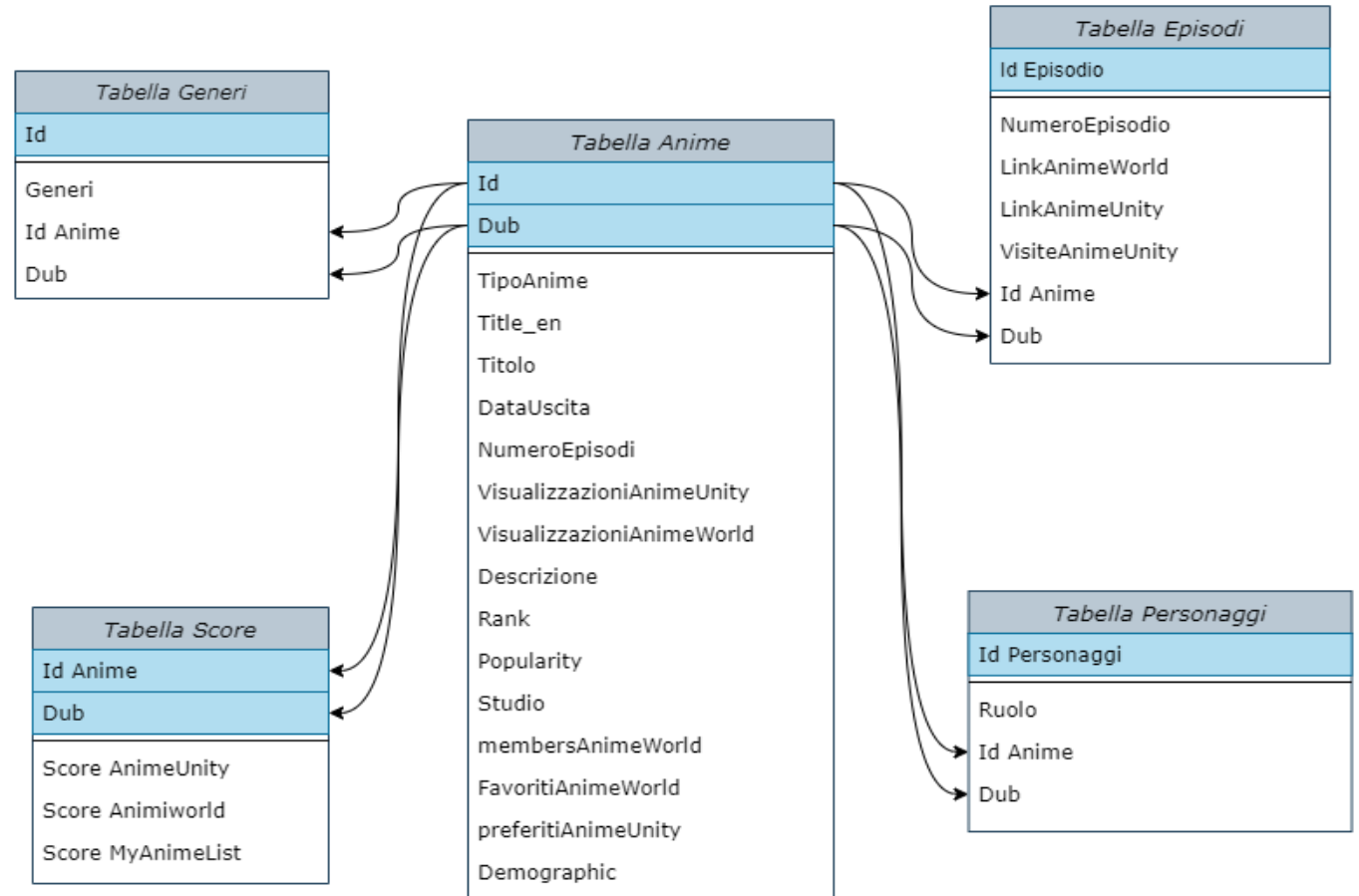
Dimensione dei dati: dal momento che abbiamo un set di dati che raccoglie gli anime dal 1966 ai primi inizi del 2023 e che in media vengono aggiunti 200 titoli all'anno, non prevediamo particolari cali di prestazioni.



Abbiamo scelto di utilizzare **MySQL** come sistema di database. Si tratta di un database relazionale open-source che gode di una vasta comunità di supporto e di una solida reputazione. Supporta anche l'utilizzo di **SQL** come linguaggio di interrogazione offrendo numerosi vantaggi. Ad esempio, consente di effettuare joints tra più tabelle per ottenere risultati complessi.

Modellazione Database

- Le istanze principali da gestire riguardano gli anime stessi, i personaggi, gli episodi, gli score e i generi per ogni titolo disponibile. L'approccio alla definizione di 5 tabelle si basa sul presupposto di avere molteplici valori per ogni istanza all'interno di ogni anime.
- La struttura del nostro database è stata progettata per semplificare le funzioni di interrogazione e offrire una grande flessibilità.
- Le caselle con sfondo blu celeste rappresentano le chiavi primarie, mentre le frecce indicano le chiavi esterne che stabiliscono le relazioni tra le tabelle.



04 →

Data Cleaning e Data Integration

Data Cleaning e Data Integration

- Data cleaning
- Data integration

```
1 import json
2 import mysql.connector
3 import os
4 from dotenv import load_dotenv
5 import re
6
7 load_dotenv()
8
9 USERNAME = os.getenv('databaseUser')
10 PASSWORD = os.getenv('databasePassword')
11 regex = r"<dd>\d+\s+(w+\s+(\d+)</dd>"
12 HOST = os.getenv('host')
13
14 mydb = mysql.connector.connect(
15     host = HOST,
16     user = USERNAME,
17     password = PASSWORD,
18     database = 'animedb2'
19 )
20 def contains_ita(string):
21     return 1 if "(ITA)" in string else 0
22
23 dataset = {}
24 with open('Database/items_listAnimeWorld.json', 'r', encoding="utf-8") as infile:
25     dataset = json.load(infile)
26
27 cursor = mydb.cursor()
28
29 for anime in dataset["records"]:
30     idAnime = anime["MyAnimeList"]
31
32     if(idAnime is None):
33         print("")
34     else:
35         idAnime = idAnime.split("/anime/",1)[1].strip()
36         print(idAnime)
37         visualizzazioni = anime["Visualizzazioni"]
38         scoreAnimeWorld = anime["Rating"]
39         titolo = anime["Title"]
40         dub = str(contains_ita(titolo))
41         descrizione = anime["Description"]
42         stagione = anime["Stagione"]
43         annoUscita = anime["Data_uscita"]
44
45         cursor.execute(f"select * from anime where id = " + str(idAnime))
46         records = cursor.fetchall()
47
48         print("idAnime = " + idAnime + ", titolo = " + titolo)
49
50         match = re.search(regex, annoUscita)
```

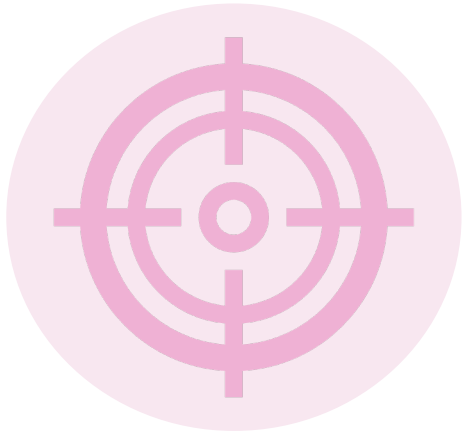
```
51 match = re.search(regex, annoUscita)
52 anno = 0
53 if match:
54     anno = match.group(1)
55
56 if(records == []):
57     sql = "INSERT INTO anime (id, dub, titolo, stagione, dataUscita, visualizzazioneAnimeWorld, descrizione) VALUES (%s, %s, %s, %s, %s, %s, %s)"
58     val = (idAnime, dub, titolo, stagione, anno, int(visualizzazioni.replace(".", "")), descrizione)
59     cursor.execute(sql, val)
60
61 else:
62     sql = "UPDATE anime SET visualizzazioneAnimeWorld = %s WHERE id = %s and dub = %s"
63     val = (int(visualizzazioni.replace(".", "")), idAnime, dub)
64     cursor.execute(sql, val)
65
66     sql = "UPDATE anime SET visualizzazioneAnimeWorld = %s, titolo=%s WHERE id = %s and dub = %s and titolo IS NULL"
67     val = (int(visualizzazioni.replace(".", "")), titolo, idAnime, dub)
68     cursor.execute(sql, val)
69
70 for episodios in anime["Link-ep"]:
71     for episodio in episodios:
72         linkAnime = episodios[episodio]
73         cursor.execute(f"select * from episodi where idAnime = " + str(idAnime) + " AND dub = " + dub + " AND numeroEpisodio = " + episodio)
74         records = cursor.fetchall()
75
76         if(records == []):
77             sql = "INSERT INTO episodi (numeroEpisodio, idAnime, dub, linkAnimeWorld) VALUES (%s, %s, %s, %s)"
78             val = (episodio, int(idAnime), dub, linkAnime)
79             cursor.execute(sql, val)
80
81         else:
82             sql = "UPDATE episodi SET linkAnimeWorld = %s WHERE idAnime = %s and dub = %s and numeroEpisodio=%s"
83             val = (linkAnime, int(idAnime), dub, episodio)
84             cursor.execute(sql, val)
85
86 cursor.execute(f"select * from score WHERE id = " + str(idAnime) + " and dub = " + dub)
87 records = cursor.fetchall()
88
89 if(records == []):
90     sql = "INSERT INTO score (id, dub, animeworld) VALUES (%s, %s, %s)"
91     val = (idAnime, dub, scoreAnimeWorld)
92     cursor.execute(sql, val)
93
94 else:
95     sql = "UPDATE score SET animeworld = %s WHERE id = %s AND dub = %s"
96     val = (scoreAnimeWorld, idAnime, dub)
97     cursor.execute(sql, val)
98
99 mydb.commit()
100 cursor.close()
101 mydb.close()
```

05



Data Quality

Data Quality



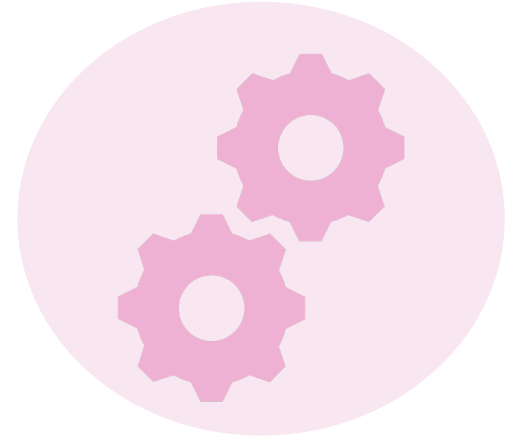
ACCURACY



COMPLETENESS



CURRENCY



CONSISTENCY

Accuracy

- ❑ Corretta rappresentazione di un fenomeno della realtà
- ❑ Fonti non ufficiali (possibili errori)
- ❑ Gestione del titolo delle serie:
 - Riportato secondo le due fonti italiane nella colonna 'titolo';
 - Riportato secondo la fonte mondiale nella colonna 'titolo_en'.
 - Applicando SequenceMatcher al dataframe si è ottenuta una corrispondenza tra le colonne pari all'86,82%.

Completeness

- ❑ Grado di copertura con il quale il fenomeno è rappresentato nel dataset
 - Calcolato a livello di singolo attributo per ogni tabella
- ❑ Per la tabella 'Anime' è pari all'89,27%, su un totale di 3.927 righe
- ❑ Per la tabella 'Episodi' è pari all'89,8%, su un totale di 67.307 righe
- ❑ Per la tabella 'Genere' è pari al 100%, su un totale di 18.552 righe
- ❑ Per la tabella 'Personaggi' è pari al 100%, su un totale di 89.990 righe
- ❑ Per la tabella 'Score' è pari al 90,38%, su un totale di 4.519 tuple.

Attributo	Nan Valori	Percentuale Nan (%)
Id	0	0,00
Dub	0	0,00
NumeroEpisodio	0	0,00
Id_Anime	0	0,00
LinkAnimeWorld	6.058	9.14
LinkAnimeUnity	20.208	30.50
VisiteAnimeUnity	20.208	30.50

Table 2: Tabella Episodi

Currency

- Misura la frequenza con la quale i dati sono aggiornati
- Nel nostro caso i dati cambiano continuamente:
 - Nuove serie;
 - Nuovi personaggi;
 - Nuovi episodi;
 - Nuove serie doppiate in italiano.
- Per assicurare la currency, il nostro database dovrebbe essere aggiornato almeno settimanalmente

Consistency

- ❑ Descrive la consistenza di diverse rappresentazioni dello stesso oggetto della realtà nel database
- ❑ Nello specifico, si sono presi in esame gli attributi più significativi della tabella 'Anime'

Attributo AnimeU	AnimeU DType	Attributo AnimeW	AnimeW DType	Attributo MAL	MAL DType
Id	integer	Id	integer	Id	integer
Dub	boolean	Dub	string	-	-
Stagione	string	Stagione	string	-	-
TipoAnime	string	TipoAnime	string	-	-
Titolo	string	Titolo	string	Title	string
Data_uscita	integer	Data di uscita	string	-	-
Plot	string	Description	string	-	-
-	-	-	-	Ranked	integer
-	-	-	-	Popularity	integer
Studio	string	-	-	-	-
Members	integer	-	-	Members	integer
Studio	string	-	-	-	-

Table 6: Tabella di Consistency

06



Analisi Risultati

Confronto top 100 AnimeUnity e top 100 AnimeWorld

Abbiamo eseguito un'operazione di join interna degli anime più visti sulle due piattaforme e abbiamo ottenuto un totale di 89 titoli in comune. Tra questi, rileviamo 15 anime che presentano l'attributo dub=1, indicando che si tratta di una versione doppiata. In generale, si osserva che la versione doppiata presenta meno visualizzazioni rispetto alla controparte in lingua originale.

Facendo altre queries scopriamo che:

- ❑ sono stati rilasciati 36 anime nel periodo compreso tra il 2019 e il 2021, con il 2019 che risulta l'anno con più titoli popolari. L'anno di rilascio degli anime dell'insieme varia invece dal 1986 al 2022, coprendo un ampio range temporale. In relazione alla stagione, l'autunno con 32 titoli risulta essere il periodo più indicato per l'uscita di una nuova serie.
- ❑ il genere Action compare 49 volte, seguito dal Fantasy con 36 e Adventure con 28 titoli. Se invece consideriamo l'insieme dei generi per ogni anime, emerge che la combinazione Action, Adventure e Fantasy è la più apprezzata dal pubblico italiano.
- ❑ lo studio David Production si distingue per aver realizzato 7 anime di successo, totalizzando 9,4 milioni di views tra le due piattaforme streaming. Tenendo conto unicamente del titolo, si tratta delle saghe di "Enen no Shouboutai" e "Jojo". Al contrario, se consideriamo lo studio con il maggiore numero di anime differenti di successo troviamo lo studio Perrot, Madhouse e TMS Entertainment ciascuno con 4 anime.

id	dub	titolo	avg_score	myanimelist	visualizzazioneAnimeUnity	visualizzazioneAnimeWorld
21	0	One Piece	9.41	8.69	35.224.643	24.550.641
1.735	0	Naruto: Shippuuden	9.34	8.26	11.059.425	11.349.667
21	1	One Piece (ITA)	9.25	8.69	11.910.145	15.971.310
1.735	1	Naruto: Shippuuden (ITA)	9.24	8.26	15.910.229	12.809.491
5.114	0	Hagane no Renkinjutsushi: Fullmetal Alchemist	9.22	9.1	343.263	1.027.205
11.061	0	Hunter x Hunter (2011)	9.18	9.04	5.421.380	7.451.319
40.776	0	Haikyuu!!: To the Top 2	9.1	8.54	1.183.836	816.915
813	1	Dragon Ball Z (ITA)	9.06	8.16	4.300.157	3.308.871
47.778	0	Kimetsu no Yaiba: Yuukaku-hen	9.06	8.8	1.356.295	1.705.933
48.583	0	Shingeki no Kyojin: The Final Season Part 2	9.05	8.77	954.920	1.455.321
235	0	Meitantei Conan	9.04	8.17	8.080.783	5.253.122
32.935	0	Haikyuu!! 3	9.04	8.78	591.842	747.532
39.551	0	Tensei shitara Slime Datta Ken 2	9.02	8.39	785.147	1.009.288
41.487	0	Tensei shitara Slime Datta Ken 2 Part 2	9	8.33	575.718	897.884
40.028	0	Shingeki no Kyojin: The Final Season	8.98	8.8	183.666	1.306.682
35.972	0	Fairy Tail: Final Series	8.98	7.57	1.016.618	1.663.218
38.883	0	Haikyuu!! to the top	8.96	8.36	974.990	1.023.431
263	0	Fighting Spirit	8.95	8.76	541.539	699.323
269	1	Bleach (ITA)	8.93	7.91	1.806.960	1.281.987
37.991	0	JoJo no Kimyou na Bouken Part 5: Ougon no Kaze	8.93	8.58	237.101	1.104.673
5.231	1	Inazuma Eleven (ITA)	8.92	7.69	2.959.245	1.577.921
20	0	Naruto	8.91	7.98	2.762.365	2.197.291
20	1	Naruto (ITA)	8.88	7.98	8.959.749	9.053.289

Confronto con MyAnimeList: Fase esplorativa

- ❑ Da una prima analisi emerge che gli anime più popolari in termini di visualizzazioni non sono necessariamente quelli che hanno ottenuto uno score medio più elevato. Questo suggerisce l'assenza di una forte associazione tra il numero di views e lo score medio ottenuto.
- ❑ Se consideriamo lo studio, la casa di produzione più prolifica dal 1986 al 2022 `e A-1 Pictures con 99 titoli rilasciati, seguita da "Studio Denn" con 99 e "J.C: Staff" con 73 anime.
- ❑ Un dato interessante `e che l'anime che ha ottenuto il punteggio più alto secondo la nostra classifica deriva da una collaborazione tra lo studio "Trigger" e lo studio "CloverWorks".Invece, se prendiamo in considerazione le case di produzione con almeno 10 titoli prodotti, "Wit Studio" registra una media di 922 mila membri per anime e uno score medio di 7.97.

id	titolo	popularity	members	ranks	myanimelist
16.498	Shingeki no Kyojin	1	3.723.767	107	8.54
5.114	Hagane no Renkinjutsushi: Fullmetal Alchemist	3	3.158.223	1	9.1
30.276	One Punch Man	4	3.041.495	128	8.5
11.757	SAO	5	2.938.798	3.051	7.2
31.964	Boku no Hero Academia	6	2.866.953	742	7.89
38.000	Demon Slayer: Kimetsu no Yaiba	7	2.782.386	122	8.51
20	Naruto	8	2.702.446	614	7.98
22.319	Tokyo Ghoul	9	2.686.161	943	7.79
11.061	Hunter x Hunter (2011)	10	2.638.223	10	9.04
32.281	Kimi no Na wa	11	2.580.898	28	8.85
25.777	Shingeki no Kyojin 2	12	2.550.857	129	8.5
33.486	Boku no Hero Academia 2	14	2.398.508	446	8.11
1.735	Naruto: Shippuuden	15	2.334.552	280	8.26

Confronto con MyAnimeList: Top 100 MyAnimeList

Nella comparazione tra la fonte internazionale e le fonti italiane prendiamo in considerazione solo gli anime sottotitolati, in modo da non avere titoli duplicati ma un risultato più coerente. Questa decisione è stata presa dopo aver verificato se i titoli, con la versione italiana, siano o meno presenti nella top 100 di MyAnimeList.

Il risultato ottenuto è una lista di **32 anime**. Le tre fonti condividono il **38,5%** dei titoli delle rispettive Top 100.

- ❑ La valutazione media degli anime è pari all'8,11% per MyAnimeList, all'8,64% per AnimeUnity e all'8,47% per AnimeWorld.
- ❑ Dallo studio "A1-Picutres" emerge la maggiore consistenza nel creare anime di successo, con 4 diversi titoli che hanno riscosso un apprezzamento a livello mondiale.
- ❑ La combinazione dei generi Azione, Avventura e Fantasy si conferma la scelta migliore per attirare una vasta platea di utenti.

id	dub	titolo	popularity	myanimelist	animeunity	animeworld
5.114	0	Hagane no Renkinjutsushi: Fullmetal Alchemist	3	9.1	9.22	8.78
38.000	0	Demon Slayer: Kimetsu no Yaiba	7	8.51	8.75	8.56
20	0	Naruto	8	7.98	8.91	8.55
11.061	0	Hunter x Hunter (2011)	10	9.04	9.18	8.71
33.486	0	Boku no Hero Academia 2	14	8.11	8.34	8.2
1.735	0	Naruto: Shippuuden	15	8.26	9.34	8.63
40.748	0	Jujutsu Kaisen	17	8.64	8.85	8.63
21	0	One Piece	20	8.69	9.41	8.38
31.240	0	Re:Zero kara Hajimeru Isekai Seikatsu	25	8.24	8.47	8.16
36.456	0	Boku no Hero Academia 3	26	8.04	8.39	8.17
22.199	0	Akame ga Kill!	29	7.47	8.25	8.18
23.755	0	Nanatsu no Taizai	31	7.67	8.23	8.2
24.833	0	Ansatsu Kyoushitsu	34	8.09	8.42	8.37
20.583	0	Haikyuu!!	36	8.44	8.88	8.64
9.919	0	Ao no Exorcist	39	7.5	8	8
269	0	Bleach	40	7.91	8.78	8.27
40.028	0	Shingeki no Kyojin: The Final Season	41	8.8	8.98	8.48
6.702	0	Fairy Tail	49	7.57	8.45	8.02

07 →

Conclusioni e Sviluppi futuri

Conclusioni e sviluppi futuri

- In conclusione, abbiamo creato un database relazionale per le serie di "anime" da tre fonti di dati ricavate da AnimeWorld, AnimeUnity e MyAnimeList. Il database così ottenuto, contiene tutti i personaggi e gli episodi delle serie, con attributi utili come i rating, lo studio di produzione e il genere, consentendoci di rispondere ai numerosi interrogativi che ci siamo posti.
- Il database può essere arricchito e ampliato ogni volta che vengono aggiornati anime, episodi o personaggi. Siamo soddisfatti che il flusso di lavoro possa essere quasi completamente automatizzato.
- Per gli sviluppi futuri, l'automazione è sicuramente interessante e ci permetterebbe di approfondire altri aspetti. La possibile aggiunta di nuovi attributi, come il nome del doppiatore o il sesso dei personaggi principali, potrebbe permetterci di effettuare confronti più complessi per la tabella personaggi, ad esempio osservando quali doppiatori compaiono in serie di successo.