Progetto Tecniche per la gestione degli Open Data

Aeroporti Italiani

Docente: Davide Taibi
A cura di: Alessandro Frenna (0652004)

Aeroporti Italiani, indice della relazione

Descrizione del progetto

Aeroporti Italiani nasce dall'idea di creare dei linked open data a partire dai i dati forniti dall'Ente Nazionale per l'Aviazione Civile (ENAC) sugli aeroporti commerciali italiani, e dai dati sul traffico messi a disposizione sia dall'Istituto Nazionale di Statistica (ISTAT) che dall'ENAC stessa. A partire da questi dati, messi assieme attraverso una pipeline di elaborazione, è stata creata la mappa degli aeroporti italiani consultabile all'indirizzo: http://u.osmfr.org/m/624143/

Dati utilizzati e licenze

I dati da cui è stata creata la base di conoscenza sugli aeroporti italiani sono:

1. Elenco aeroporti italiani certificati - aggiornamento al 23 marzo 2021 (CSV) fornito dall'ENAC tramite il portale open data del Ministero delle Infrastrutture e Trasporti e fornito sotto licenza Creative Commons Attribution 4.0;
2. Trasporto Aereo (CSV) fornito dall'ISTAT tramite il suo portale open data, fornita con licenza Creative Commons License – Attribution – 3.0 come qui riportato;
3. Dati di Traffico 2020 (PDF) forniti dall'ENAC sul suo portale web ma di cui non sono riportate informazioni sul tipo di licenza.

Le licenze CC BY 4.0/3.0 usate dai dataset proposti consentono l'utilizzo dei dati anche per fini commerciali a patto di citare la fonte dei dati utilizzati (come sopra) e a condizione che non vengano applicate delle restrizioni o che vengano fornite garanzie sui dataset rilasciati.

Note di rilascio

I dataset prodotti sono da intendersi sotto licenza Creative Commons Attribution 4.0 e sono, pertanto, riutilizzabili da chiunque, anche per fini commerciali.
La mappa degli aeroporti italiani presente al seguente link http://u.osmfr.org/m/624143/, è rilasciata sotto licenza Open Data Commons Open Database License (ODbL)

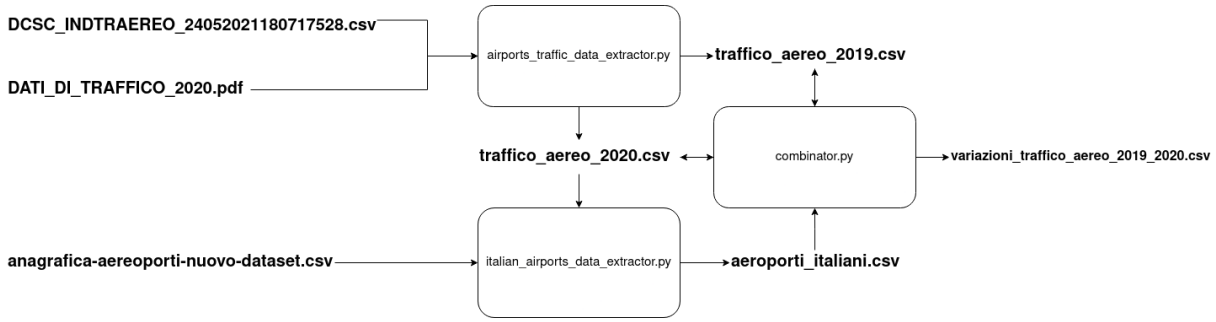
Pipeline di elaborazione

Ovviamente, i dati utilizzati hanno dovuto subire un processo di elaborazione e di “pulizia” prima di essere trasformati in linked open data. La pipeline di elaborazione dei file è stata scritta in linguaggio python ed il repository GIT del progetto è il seguente: https://github.com/alessandrofrenna/aeroporti_italiani

I file che sono stati elaborati, scaricati dai portali dell’ENAC e dell’ISTAT sono denominati come segue:

- ***anagrafica-aereoporti-nuovo-dataset.csv***;
- ***DCSC_INDTRAEREO_24052021180717528.csv***;
- ***DATI_DI_TRAFFICO_2020.pdf***.

Per l’elaborazione dei file in formato CSV è stata utilizzata la libreria Pandas al posto del modulo csv della libreria standard di python per le funzionalità che essa ha da offrire.
Per quanto riguarda, invece il file PDF sui dati di traffico è stata utilizzata la libreria pdfplumber per estrarre le tabelle relative ai report sui dati del traffico aereo, contenute nelle pagine 31 e 33.
La libreria pdfplumber è stata scelta poiché, tra le molteplici alternative trovate, è risultata sia quella con meno dipendenze necessarie che quella più semplice ed immediata da utilizzare.
Il file “anagrafica-aeroporti-nuovo-dataset.csv”, non essendo codificato in UTF-8, sul sistema su cui sono stati elaborati i dati (Ubuntu 20.04 LTS) presenta dei simboli sconosciuti in corrispondenza delle lettere accentate e di altri caratteri speciali, pertanto, all’interno della pipeline di elaborazione sono stati apportati, laddove necessario, dei check specifici in corrispondenza delle righe del file CSV in base alle necessità.
Come si può vedere sul repository GIT sopra linkato, all’interno della directory “**src**” è presente il package “**pipeline**” che al suo interno contiene i sorgenti, scritti nel linguaggio python, che compongono la pipeline di elaborazione.
Di seguito viene proposto lo schema della pipeline realizzata:

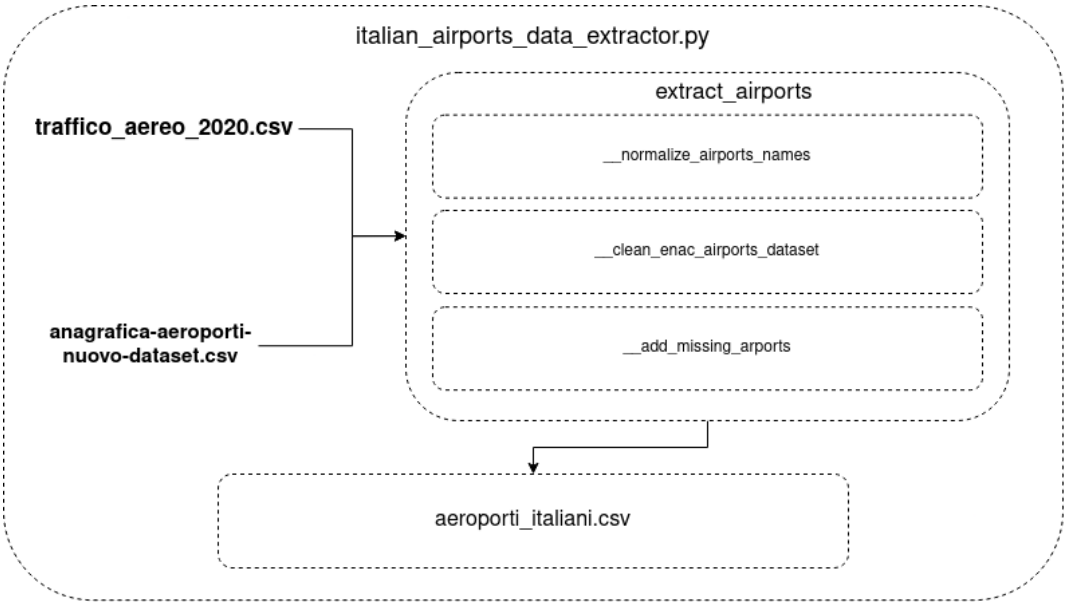


Analizzeremo adesso, in dettaglio, ogni blocco presente nell’immagine sopra presentata.

Estrazione dati sul traffico aeroportuale

Il file "**src/pipeline/airports_traffic_data.py**" viene utilizzato nella pipeline di elaborazione per generare i file csv "**data/traffico_aereo_2019.csv**" e "**data/traffico_aereo_2020.csv**" che contengono le informazioni sul traffico aereo per gli anni 2019 e 2020, rispettivamente, relativi agli aeroporti commerciali italiani. All'interno dei due file ottenuti i dati sul traffico sono raggruppati per categoria di traffico (merci e posta, passeggeri, voli commerciali) e per origine (nazionale, internazionale).

Estrazione dati sugli aeroporti italiani

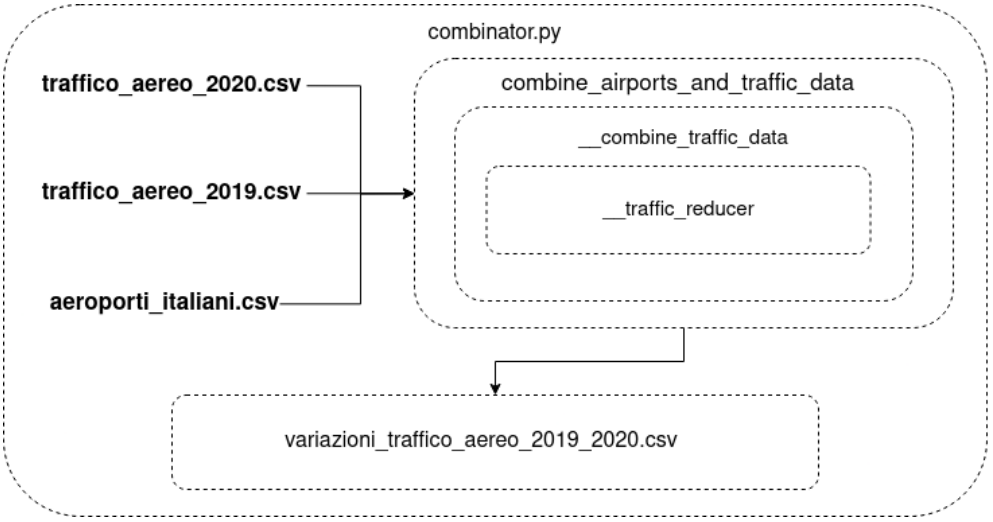


Il file "**src/pipeline/italian_airports_data_extractor.py**" è usato nella pipeline di elaborazione per ottenere il file "**data/aeroporti_italiani.csv**" che contiene la lista degli aeroporti commerciali italiani. La funzione "*extract_airport*" viene usata per:

1. estrarre il nome degli aeroporti italiani che sono memorizzati all'interno del file "traffico_aereo_2020.csv" nella colonna che ha intestazione "Aeroporto";

2. normalizzare i nomi degli aeroporti contenuti nel file “anagrafica-aeroporti-nuovo-dataset.csv” attraverso i dati estratti al punto precedente tramite la funzione “*__normalize_airports_names*”;
3. ripulire i dati contenuti nel file “anagrafica-aeroporti-nuovo-dataset.csv” in modo da ottenere dei dati meglio strutturati. La funzione “*__clean_enac_airports_dataset*” permette di analizzare il DataFrame Pandas ottenuto dalla lettura del csv sugli aeroporti e di estrarre i dati che, verranno poi inseriti all’interno del file elaborato. Quello che si può notare analizzando il codice sorgente è che, in tale funzione, viene utilizzato il servizio Nominatim.Reverse (mediante geopy) per ottenere, a partire dalle coordinate geografiche (latitudine e longitudine), i dati territoriali associati al luogo in cui si trova l’aeroporto, come regione, comune, provincia (l’indirizzo è preso dal csv di partenza). Vengono anche effettuati dei check sui valori che presentano dei caratteri non riconosciuti (e.g. accenti). Altra funzionalità presente nella funzione “*__clean_enac_airports_dataset*” è quella dell’estrazione dei dati relativi alle persone a cui sono intitolati gli aeroporti, fatta attraverso la funzione “*__extract_airports_eponyms*” che, analizza i valori contenuti nella colonna denominata “NOME COMMERCIALE” all’interno del DataFrame degli aeroporti ed estrae da ta tali valori i nomi delle persone ad essi associati;
4. aggiungere eventuali aeroporti mancanti (solamente l’aeroporto di Pantelleria) tramite la funzione “*__add_missing_airports*”.

Estrazione dati sul riepilogo del traffico



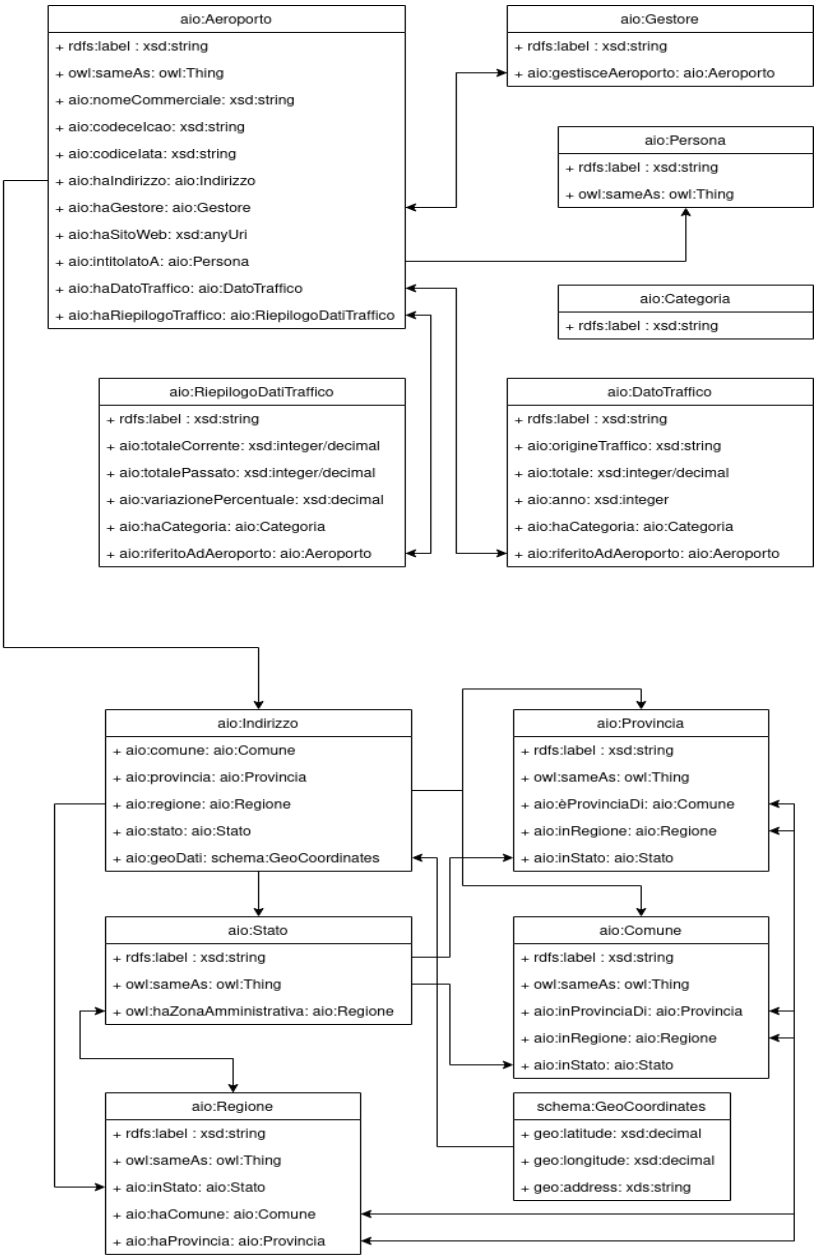
Il file “**src/pipeline/combinator.py**” ha il compito di combinare i dati sul traffico ottenuti precedentemente in un unico file denominato “**variazioni_traffico_aereo_2019_2020.csv**”, che contiene al suo interno i dati sul traffico aereo totale (nazionale e internazionale) raggruppati per categoria di traffico sia per l’anno 2019 che per il 2020.
Per ogni categoria di traffico viene poi calcolata la variazione percentuale tra il valore del 2019 e il valore del 2020 attraverso la formula:

$$\left(\frac{valore2020 - valore2019}{valore2019}\right) \cdot 100$$

Vengono poi sostituiti i nomi contenuti nella colonna con intestazione “Aeroporto” dei file “traffico_aereo_2019.csv” e “traffico_aereo_2020.csv”, con i codici ICAO degli aeroporti estratti in precedenza.

Creazione dell’ontologia

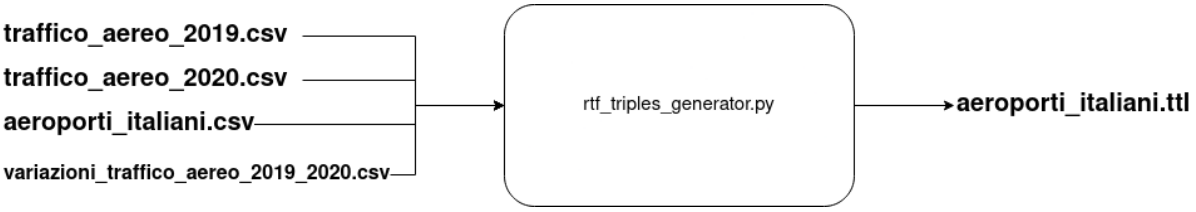
Tramite il software opensource Protege è stata poi creata l’ontologia OWL (Web Ontology Language) associata ai dati estratti. E’ stato registrato su purl.org, per comodità, il dominio “*http://purl.org/net/aeroporti_italiani*” per far sì che le risorse presenti all’interno degli open linked data che genereremo abbiano sempre un URI consistente.
L’ontologia è consultabile all’indirizzo http://purl.org/net/aeroporti_italiani/ontologia.owl.
Di seguito viene illustrato lo schema delle classi, e delle object/datatype property presenti nell’ontologia.



Il namespace “aio” sta per “aeroporti italiani ontologia” ed è stato scelto come prefisso delle nostre classi, object properties e datatype properties.

Creazione file RDF e interlinking con DbPedia e Wikidata

Una volta definita l’ontologia, la pipeline di elaborazione è stata completata creando il file RDF/Turtle a partire dai file estratti precedentemente.



Il file “**src/pipeline/rdf_triples_generator.py**” serve per produrre il file “**aeroporti_italiani.ttl**” che sarà, alla fine della pipeline, il file RDF/Turtle che conterrà la nostra base di conoscenza. Tale file RDF/Turtle è creato tramite la libreria python rdflib.
All’interno di questo file contenente codice in linguaggio python, i dati precedentemente estratti sono collegati con i dati di DbPedia e di Wikidata.
Da DbPedia le risorse collegate sono relative alle risorse di tipo **Comune**, **Provincia** e **Regione** (oltre alla risorsa che rappresenta lo **Stato** italiano) presenti nella nostra base di conoscenza.
Da Wikidata sono state collegate le risorse equivalenti a:

- risorse di tipo **Aeroporto**;
- risorse di tipo **Persona**.

Tali risorse sono state estratte da Wikidata attraverso query “**SPARQL**” mediante la libreria sparqlwrapper.
La query utilizzata per estrarre gli aeroporti è visibile qui a fianco. Come si può vedere gli aeroporti sono stati selezionati insieme al relativo codice ICAO e all’informazione sul loro sito web. Il codice ICAO serve per associare successivamente la risorsa Wikidata all’aeroporto corrispondente della nostra base di conoscenza. Il dato relativo al sito web è stato inserito negli aeroporti estratti come dato aggiuntivo.

```
query = """
    SELECT DISTINCT ?airport ?icao ?website
    WHERE {
        ?airport wdt:P239 ?icao;
        wdt:P856 ?website.
        filter(strstarts(?icao,"LI"))
    }
"""
```

La seguente query è stata invece usata per estrarre i dati relativi alle persone a cui sono intitolati gli aeroporti.

```
query = """
    SELECT DISTINCT ?item ?itemLabel WHERE {
        ?item wdt:P31 wd:Q5.
        ?item ?label "$1"@it.
        SERVICE wikibase:label { bd:serviceParam wikibase:language "it".
    }
```

Per ogni eponimo presente ed associato ad un aeroporto, è stata eseguita la query qui sopra, con la quale è stato effettuato un match tra il valore della label associata a risorse di tipo "**human**" (wd:Q5) su Wikidata, ed il nome della persona a cui è intitolato l'aeroporto ($1). E' risultato impossibile utilizzare query che utilizzassero metodi e filtri più efficienti per ottenere fuori queste informazioni da Wikidata senza che queste andassero in timeout. Probabilmente il problema è causato dalla ricerca per nome completo. I risultati ottenuti dall'esecuzione delle query, laddove sono state ottenute più risorse corrispondenti alla chiave di ricerca, sono stati ulteriormente filtrati per arrivare alla corretta corrispondenza.

Di seguito si riporta, infine, la struttura degli URI utilizzati per ogni classe definita nell'ontologia:

1. **Aeroporto**: http://purl.org/net/aeroporti_italiani/risorse/aeroporti/CODICE_ICAO_AP;
2. **Comune**, **Provincia**, **Regione**, **Stato**: http://purl.org/net/aeroporti_italiani/risorse/luoghi/NOME_LUOGO;
3. **schema:GeoCoordinates**: http://purl.org/net/aeroporti_italiani/risorse/geo/CODICE_ICAO_AP_-Geo;
4. **Indirizzo**: http://purl.org/net/aeroporti_italiani/risorse/indirizzi/CODICE_ICAO_AP_-Indirizzo;
5. **Gestore**: http://purl.org/net/aeroporti_italiani/risorse/gestori/CODICE_ICAO_AP_-Gestore;
6. **Persona**: http://purl.org/net/aeroporti_italiani/risorse/persone/NOME_COGNOME;
7. **Categoria**: http://purl.org/net/aeroporti_italiani/risorse/categorie_traffico/CATEGORIA;
8. **DatoTraffico**: http://purl.org/net/aeroporti_italiani/risorse/dati_traffico/ICAO_CATEGORIA_ORIGINE_ANNO;
9. **RiepilogoDatiTraffico**: http://purl.org/net/aeroporti_italiani/risorse/riepiloghi_traffico/ICAO_CATEGORIA-2020-2019;