unibz
Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

Project Report

Data Profiling and Data Integration of Italian Municipalities

Bilal Mahmood (17369)

Raghavendra Prasad Akshy Sripad (18499)

Introduction and domain description

Domain

Our domain of interest includes data about 3 different municipalities in Italy, provided by 3 different sources, where the main goal of the project was to integrate them and answer policy queries on all of them. In addition, another goal of the project was to build a Python web application, where the user can interact with the underlying integrated data, and get details about all the communes present in the sources.

The data comprised about three different provinces in Italy: Avellino, Salerno, and Caserta and all the communes located inside them. A lot of information was available for each of the communes, ranging from the number of inhabitants, the number of companies, to its history, to the number of holiday events to its geographical coordinates, to name a few.

As mentioned, the main motivation for the project was to integrate the 3 sources, answer a few of the following queries and also wrap these questions around a web application that the user can utilize to answer a few of the policy questions mentioned below as well as know more facts about the communes.

- Which commune is the most developed in terms of schools and companies?
- Which is the most populous commune?
- How are the number of schools and the number of companies related?
- Which region has the least number of saints?
- Do latitude and Longitude have a relation to the number of companies?

In order to achieve this goal, we used the ontology-based integration system to combine the sources. We used several tools, ranging from the data profiling tools, such as pandas-profiling and metonome to first clean and normalize the data sources and export them into the PostgreSQL database management system to the Protégé and Ontop to define ontologies and map the ontologies to the underlying data sources. And finally, we created the SPARQL endpoint, using the Ontop-CLI tool, and queried it with Python to answer the policy questions and derive a web application built using Streamlit, to provide the user with a friendly UI to access the integrated data sources.

Data source description

We were provided with 3 raw CSV files for each of the 3 different provinces: Avellino, Salerno, and Caserta. The below table shows the number of rows in it, and the column names in each source.

Data Sources	Data Dimensions	Data Type
Avellino Municipalities	118x24	csv
Caserta Municipalities	104x24	csv
Salerno Municipalities	158x24	csv

Dimensions of the data files

Avellino Municipalities	Caserta Municipalities	Salerno Municipalities
1 Nome Comune 2 Numero di abitanti 3 Sindaco 4 Nome degli abitanti 5 Frazioni 6 Codice Postale 7 Superficie 8 Numero di frazioni 9 Etimologia del nome 10 Storia 11 Patrono 12 Giorno festivo 13 Epoca di fondazione 14 Anno di fondazione 15 Secolo di fondazione 16 Numero di punti di interesse 17 Numero di aziende 18 Numero di scuole 19 Percorsi 20 Eventi 21 Geolocalizzazione 22 Immagine 23 Sitografia 24 Bibliografia	1 Nome Comune 2 Numero di abitanti 3 Sindaco 4 Nome degli abitanti 5 Frazioni 6 Codice Postale 7 Superficie in KM 8 Numero di frazioni 9 Etimologia del nome 10 Storia 11 Patrono 12 Giorno festivo 13 Epoca di fondazione 14 Anno di fondazione 15 Secolo di fondazione 16 Numero di punti di interesse 17 Numero di aziende 18 Numero di scuole 19 Percorsi 20 Eventi 21 Geolocalizzazione 22 Immagine 23 Sitografia 24 Bibliografia	1 Nome Comune 2 Numero di abitanti 3 Sindaco 4 Nome degli abitanti 5 Frazioni 6 Codice Postale 7 Superficie 8 Numero di frazioni 9 Etimologia del nome 10 Storia 11 Patrono 12 Giorno festivo 13 Epoca di fondazione 14 Anno di fondazione 15 Secolo di fondazione 16 Numero di punti di interesse 17 Numero di aziende 18 Numero di scuole 19 Percorsi 20 Eventi 21 Geolocalizzazione 22 Immagine 23 Sitografia 24 Bibliografia

Attributes of the data sources

Data Profiling

In order to better understand the data sources we explored several data profiling tools to summarize the data sources, clean them, and assist us in bringing the 3 tables into BCNF form. The below table highlights the use of each algorithm and tool.

Algorithm and tools	Use	Purpose
Pandas-Profiling	Statistics, duplication, Null value counts	Exploration
SCDP	Statistics, duplication, Null value counts	Exploration
DUCC	Unique column combinations	Key Discovery
TANE	Functional dependency discovery	Normalization
NORMALIZE	Boyce-Codd-Normal Form Check	Normalization

Data profiling tools and their use

Data Cleaning

The first step we took was to use the pandas-profiling tool to clean the data, focusing on the policy questions goal and producing the clean data set for each of the 3 sources. We focused on removing duplicate rows, fixing the data types, standardizing the attributes such as the holiday dates, correcting mistakes due to different standards, confusion between "," and ".", for example, in the latitude and longitude fields, and removing columns that had a lot of missing values.

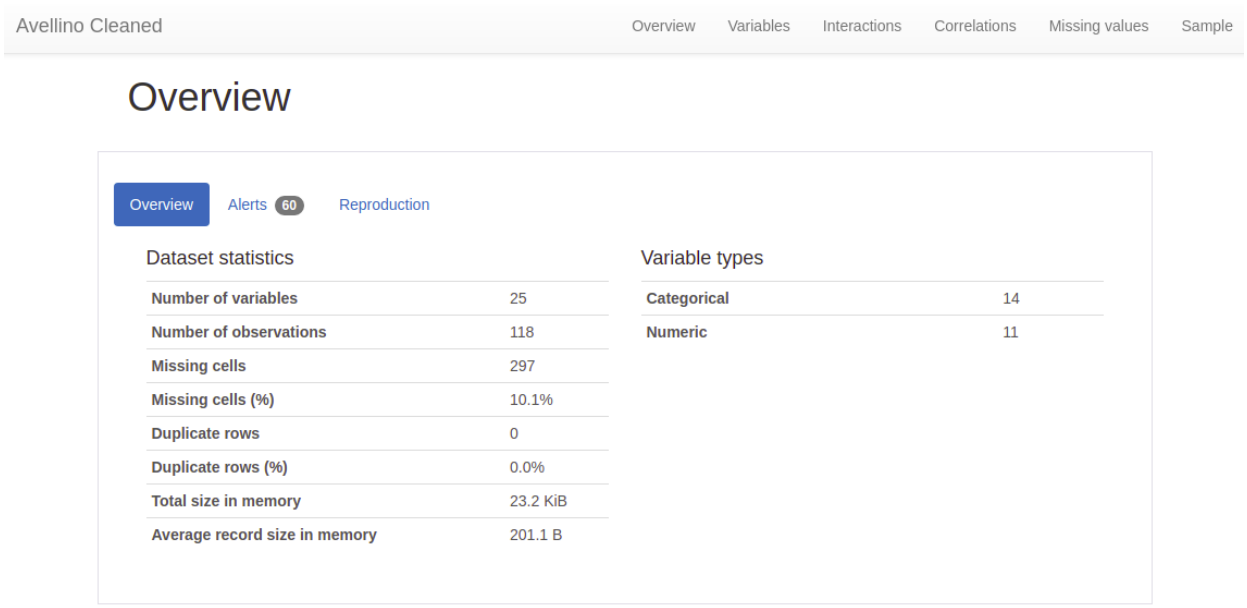
Below are the actions we took for each source, cleaning and preparing them for normalization::

Cleaning steps for Avellino Municipalities	Cleaning steps for Caserta Municipalities	Cleaning steps for Salerno Municipalities
• Frazioni "Non ha frazioni" and "Informazione assente" into NULL values • Numero di frazioni Informazione assente into NULL • Splitted Giorno festivo and break it into day and	• Frazioni "Non ha frazioni" and "Informazione assente" into NULL values • Codice Postale standardization, Castel Morrone had a missing digit 8020 →81020	• Fixed Numero di abitanti "2 268" → 2268 • Frazioni "Non ha frazioni" and "Informazione assente" into NULL values • Fixed 84050, 84040 Codice Postale for Ispani

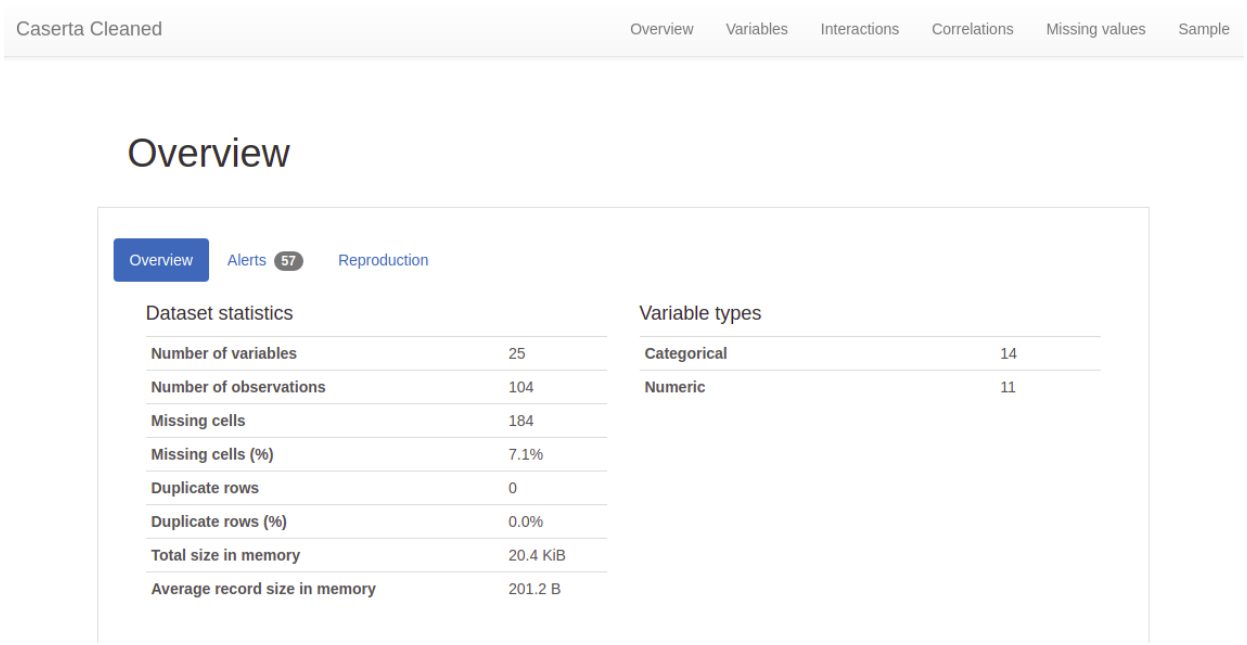
month and standardized the date. For example, 19 Settembre e Venerdì di Passione (precedente alla Domenica delle Palme) to 19 Settembre. Removed multiple dates, considering just the occurrence of the first date. Also removed date dates like lunedì dopo Pentecoste date ● Changed Epoca di fondazione “Informazione assente” into NULL ● Fixed Secolo di fondazione “Informazione assente” into NULL, and fixing values. For example, removed a.C at the ends and standardized the value IX - X into IX ● Changed Anno di fondazione “Informazione assente” into NULL and removed removing a.C at the end ● Broke Geolocalizzazione into latitude and longitude and fixing issues, for example, 40,950003 14,757562 → 40.950003, 14.757562 ● Drop Bibliografia because of all missing values	● Fixed few errors in Superficie in KM. 48,6 → 48.6 ● Giorno festivo and break into day and month and standardized the date. Eg Seconda domenica di ottobre → NULL, fixing errors Martedì in Albis → NULL, and 16 agosto, 11 novembre, 29 luglio → 16 Agosto ● Changed Anno di fondazione “Informazione assente” into NULL and removed removing a.C at the end ● Fixed Secolo di fondazione “Informazione assente” into NULL, and fixing values ● Broke Geolocalizzazione into latitude and longitude ● Drop Bibliografia because of all missing values	to 84050, Controne 840202 → 84020 ● Giorno festivo and break into day and month and standardized the date. And standardizing them. Removing multiple dates and for which we cannot know exactly day and month ● Changed Anno di fondazione “Informazione assente” into NULL and removed removing a.C at the end ● Fixed Secolo di fondazione “Informazione assente” into NULL, and fixing values ● Fixed Numero di aziende e Numero di scuole “Informazione assente” into NULL ● Broke Geolocalizzazione into latitude and longitude ● Drop Bibliografia because of all missing values

After performing above preprocessing steps, we produced the 3 cleaned data sets, one for each municipality.

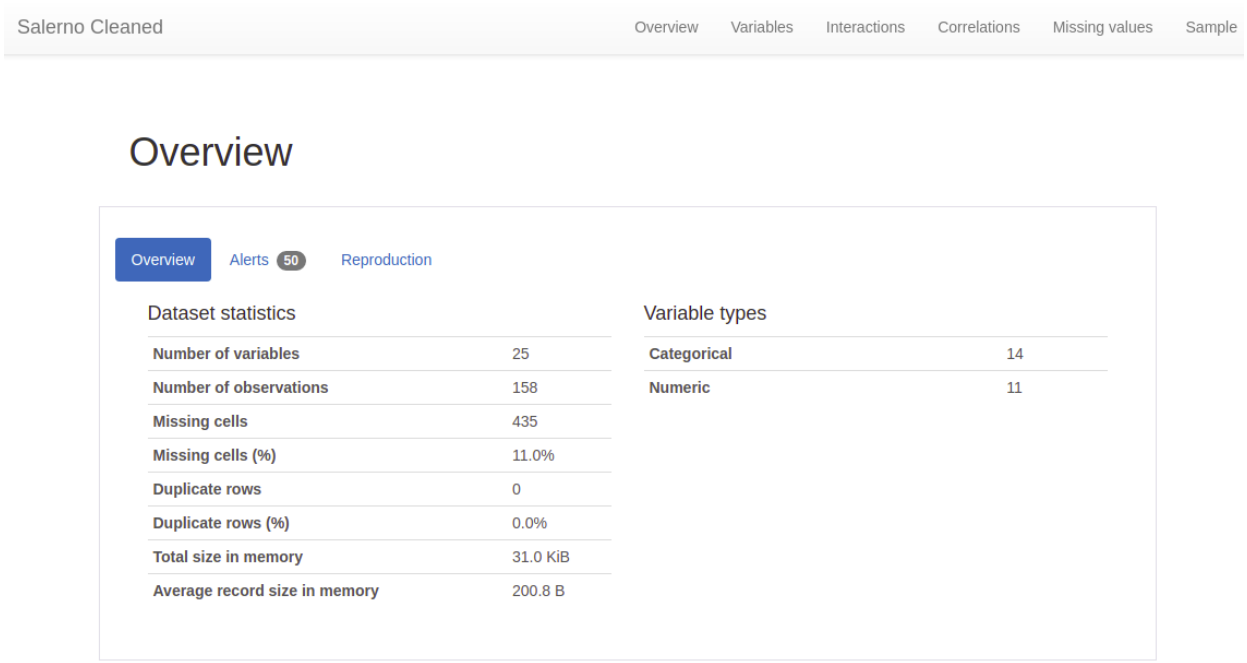
The below figures show the summaries of the cleaned data sets. They depict the correct number of variable types, correct number of missing values and no duplication in the cleaned data



Cleaned Avellino Data set



Cleaned Caserta Data set



Salerno Data set

Normalized ER Model (BC-Normal form)

To avoid data redundancy, all the data sources were normalized using domain knowledge and the Metanome FD discovery and normalization tools to make sure they were in the normalized form. The goal was to make sure that all the FDs were either trivial or their dependence was on the superkey of the table, with the goal that the ER could easily lend itself to the Data integration part of the project, where we needed to convert the data sources into a knowledge graph. Thus we came up with the below ER model for the raw data files we were provided.

Boyce Codd Normal Form (BCNF)

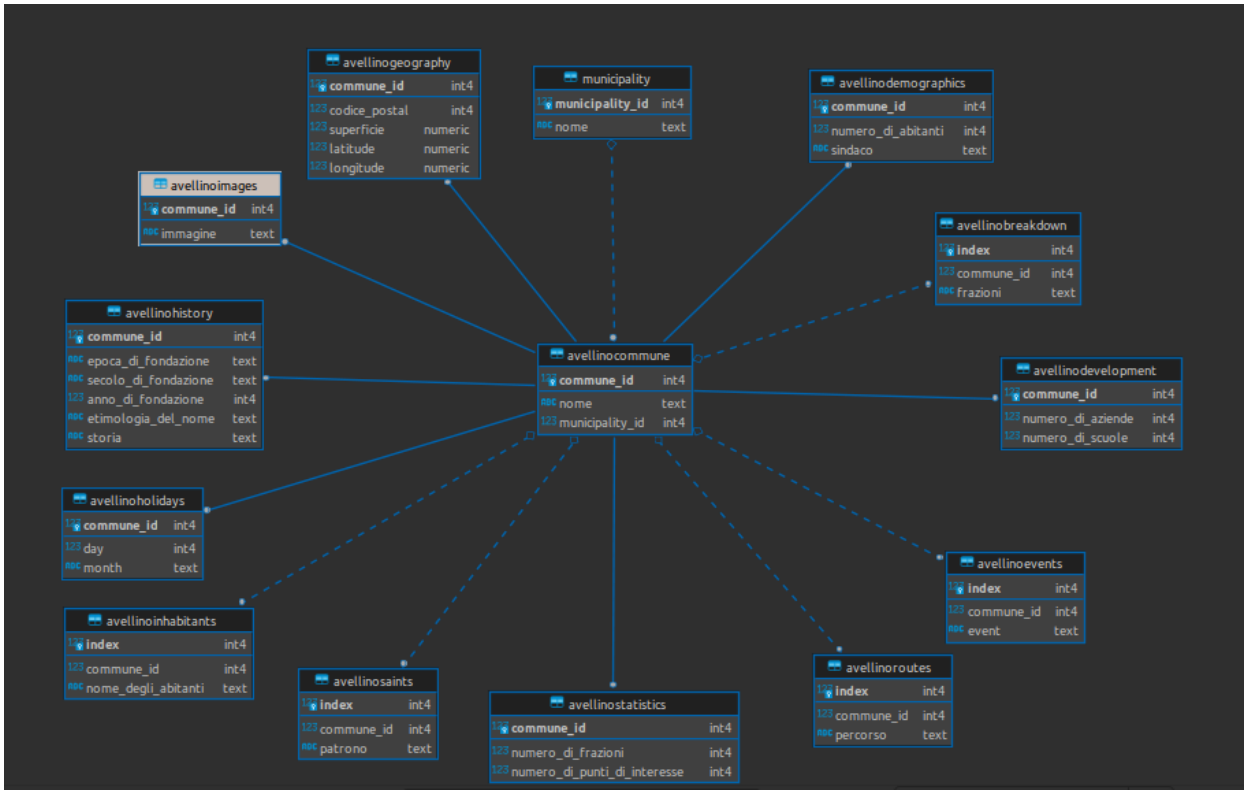
A schema is in BNCF if the following conditions hold

- For all FDs $X \rightarrow A$ whose attributes are in the **same table**
  - either $A \in X$, that is: **FD is trivial**
  - or $X \rightarrow R \setminus X$, that is: **LHS is key or superkey** of table
- This must apply to **given and inferred** FDs

Ref: Lecture slides on Duplicate Detection, Similarities 102/200

We focused on domain-oriented ER modeling, focusing on ER model that would assist in intuitive ontology-based integration, when breaking down the tables and verifying that the tables we came up with were indeed normalized, using NORMALIZE algorithm available in Metanome.

Show ER tables



Putting normalized data into PostgreSQL database for respected data source

Data Source 1: Avellino

Data Sources	UCC	NORMALIZE
Commune	2	Yes
Statistics	1	Yes
Saints	1	Yes
Breakdown	1	Yes
Routes	1	Yes
Events	0	Yes
Holidays	1	Yes
History	2	Yes
Development	1	Yes
Inhabitants	1	Yes
Demography	2	Yes
Geography	3	Yes

Data Source 2: Caserta

Data Sources	UCC	NORMALIZE
Commune	2	Yes
Statistics	1	Yes
Saints	1	Yes
Breakdown	1	Yes
Routes	0	Yes
Events	0	Yes
Holidays	1	Yes

History	3	Yes
Development	1	Yes
Inhabitants	1	Yes
Demography	3	Yes
Geography	6	No*

Data Source 3: Salerno

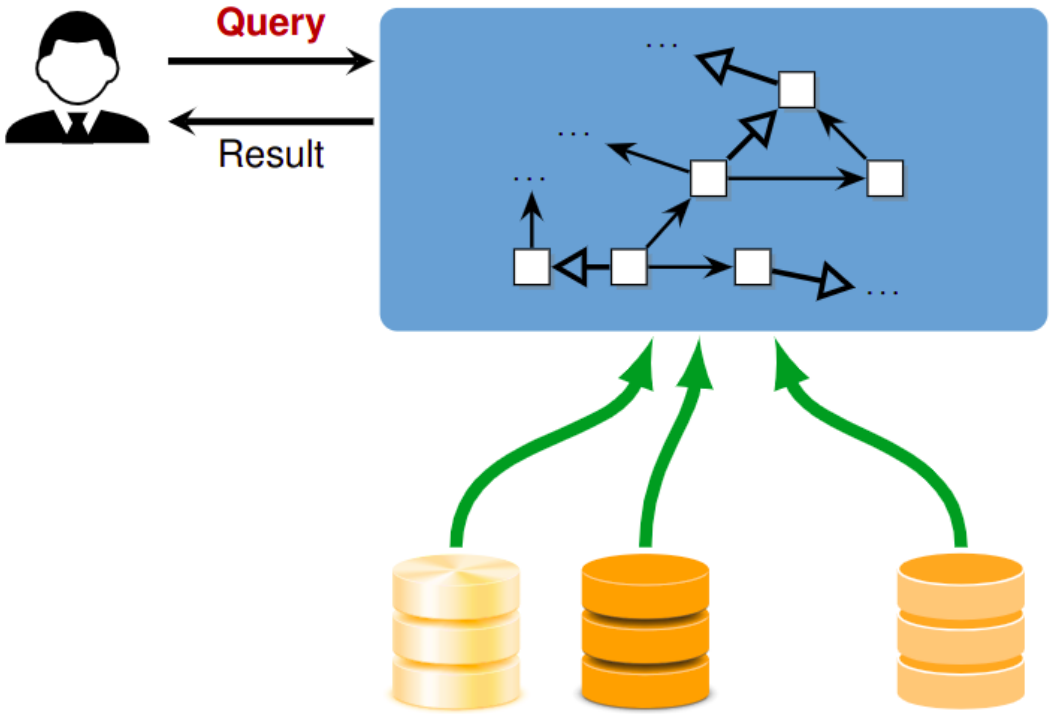
Data Sources	UCC	NORMALIZE
Commune	2	Yes
Statistics	1	Yes
Saints	1	Yes
Breakdown	1	Yes
Routes	0	Yes
Events	1	Yes
Holidays	1	Yes
History	4	Yes
Development	1	Yes
Inhabitants	1	Yes
Demography	2	Yes
Geography	5	No*

** Spurious FDs discovered between latitude and longitude features

Based on the above verification steps, we found that events, and routes, did not have a primary key, hence we added a surrogate primary key, the index column, to the tables. Furthermore, we found that the geography table was not in the normalized form and we hypothesize that it is due to the spurious FDs used by the metaonome algorithm. For example, discovering the FD between latitude and longitude, when we expect there must be no FD, as the columns are floats.

Data Integration

The main approach we used for integration was ontology based data integration where we defined the ontology, and mapped it to the underlying three data sources we had. This ontology was then queried using SPARQL, to answer the needs of the end user. The below diagram is the architecture of the data integration system.

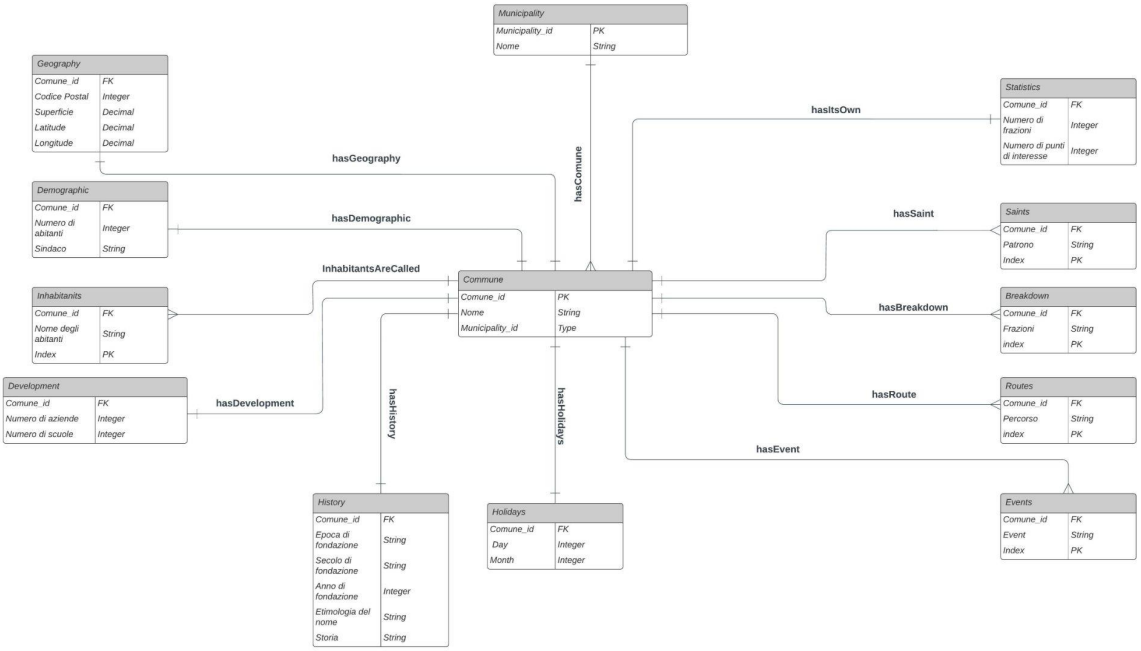


Slides: Part 4 page 4

Ontology/Mediated Schema UML diagram

To visualize an OWL2 QL ontology, we designed a UML diagram.

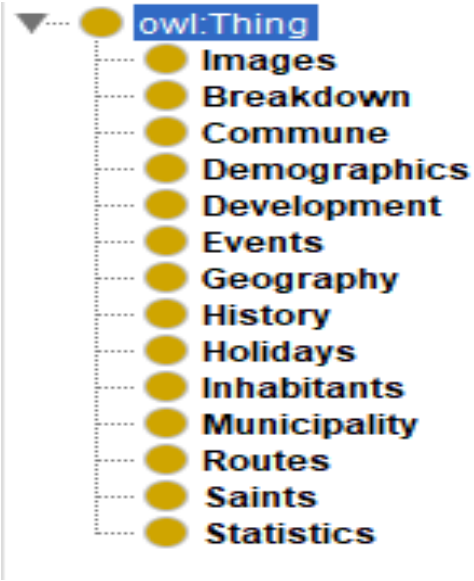
UML Model for the Data Sources



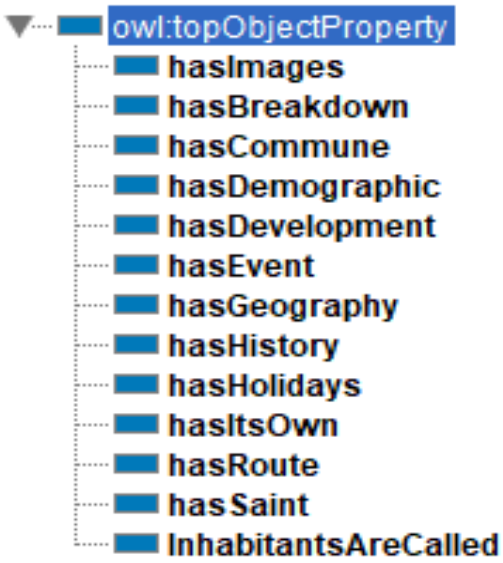
Ontology is defined as “a formal description of all the entities of a domain and the relations existing between these entities” [1]. It enables users to formulate their information needs which are then automatically translated into a query over the data sources.

The ontology was created in the Ontop system: Ontop is an open-source OBDA (Ontology Based 4 Data Access) framework released under the Apache license, developed at the Free University of Bozen-Bolzano. [2]

Classes: allow us to structure the domain of interest. We defined the most general concepts in the domain.



Object property: Classes are connected with other classes with the help of object properties.

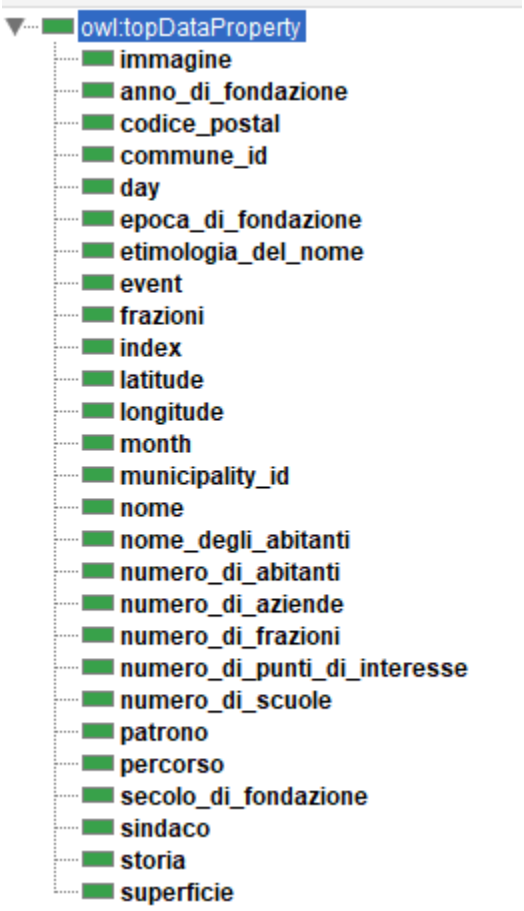


All the object properties, their domain and range is listed below:

Domain	**Object property**	**Range**
Commune	hasDemographic	Demographics
Commune	hasHistory	History
Commune	hasBreakdown	Breakdown
Commune	hasEvent	Events

Municipality	hasCommune	Commune
Commune	hasImages	Images
Commune	hasGeography	Geography
Commune	hasRoute	Routes
Commune	hasHolidays	Holidays
Commune	hasSaint	Saints
Commune	hasDevelopment	Development
Commune	InhabitantsAreCalled	Inhabitant
Commune	hasItsOwn	Statistics

Data property: connect classes with data types or literals.



All the data properties, their domain and range:

Data Property	**Domain**	**Range**
commune_id	Holidays	xsd:integer
patrono	Saints	xsd:string
storia	History	xsd:string
event	Events	xsd:string
longitude	Geography	xsd:decimal
epoca_di_fondazione	History	xsd:string
commune_id	Statistics	xsd:integer
nome_degli_abitanti	Inhabitants	xsd:string
anno_di_fondazione	History	xsd:integer
percorso	Routes	xsd:string
commune_id	Inhabitants	xsd:integer
month	Holidays	xsd:string
sindaco	Demographics	xsd:string
numero_di_scuole	Development	xsd:integer
numero_di_aziende	Development	xsd:integer
latitude	Geography	xsd:decimal
index	Breakdown	xsd:integer
superficie	Geography	xsd:decimal
commune_id	Routes	xsd:integer
secolo_di_fondazione	History	xsd:string
numero_di_abitanti	Development	xsd:integer
commune_id	Saints	xsd:integer

etimologia_del_nome	History	xsd:string
commune_id	Events	xsd:integer
frazioni	Breakdown	xsd:string
commune_id	Commune	xsd:integer
commune_id	Geography	xsd:integer
index	Events	xsd:integer
commune_id	Breakdown	xsd:integer
index	Saints	xsd:integer
commune_id	Images	xsd:integer
immagine	Images	xsd:integer
commune_id	Demographics	xsd:integer
index	Routes	xsd:integer
commune_id	Development	xsd:integer
nome	Municipality	xsd:string
numero_di_frazioni	Statistics	xsd:integer
commune_id	History	xsd:integer
numero_di_punti_di_interesse	Statistics	xsd:integer
commune_id	Municipality	xsd:integer
codice_postal	Geography	xsd:decimal
day	Holidays	xsd:integer
municipality_id	Municipality	xsd:integer
nome	Commune	xsd:string

Mappings

They are part of Target (Triple map) mappings. The source mappings are SQL queries that populate the table (select * from tablename).

We defined a total 79 mappings, connecting all the classes in the ontology we defined along with their data properties and object properties with the 3 database sources we had. Defined below are few of these mappings

MAPID-Municipality

```
:Municipality/{municipality_id} a :Municipality ; :nome {nome} .
select * from Municipality;
```

MAPID-Geography-av

```
:Geography/av/{commune_id} a :Geography ; :codice_postal {codice_postal} ; :superficie {superficie} ; :latitude {latitude} ; :longitude
select * from avellinogeography;
```

MAPID-Geography-ca

```
:Geography/ca/{commune_id} a :Geography ; :codice_postal {codice_postal} ; :superficie {superficie} ; :latitude {latitude} ; :longitude
select * from casertageography;
```

MAPID-Geography-sa

```
:Geography/sa/{commune_id} a :Geography ; :codice_postal {codice_postal} ; :superficie {superficie} ; :latitude {latitude} ; :longitude
select * from salernogeography;
```

MAPID-Routes-av

```
:Routes/av/{index} a :Routes ; :percorso {percorso} .
select * from avellinoroutes;
```

MAPID-Routes-ca

```
:Routes/ca/{index} a :Routes ; :percorso {percorso} .
select * from casertaroutes;
```

MAPID-Routes-sa

```
:Routes/sa/{index} a :Routes ; :percorso {percorso} .
select * from salernoroutes;
```

MAPID-Commune-av

```
:Commune/av/{commune_id} a :Commune ; :nome {nome} .
select * from avellinocommune;
```

MAPID-Commune-ca

```
:Commune/ca/{commune_id} a :Commune ; :nome {nome} .
select * from casertacommune;
```

MAPID-Commune-sa

```
:Commune/sa/{commune_id} a :Commune ; :nome {nome} .
select * from salernocommune;
```

MAPID-Commune-Routes-av

```
:Commune/av/{commune_id} :hasRoute :Routes/av/{index} .
select * from avellinoroutes;
```

MAPID-Commune-Routes-ca

```
:Commune/ca/{commune_id} :hasRoute :Routes/ca/{index} .
select * from casertaroutes;
```

MAPID-Commune-Routes-sa

```
:Commune/sa/{commune_id} :hasRoute :Routes/sa/{index} .
select * from salernoroutes;
```

MAPID-Commune-Statistics-av

```
:Commune/av/{commune_id} :hasItsOwn :Statistics/av/{commune_id} .
select * from avellinostatistics;
```

MAPID-Commune-Demographics-av

```
:Commune/av/{commune_id} :hasDemographic :Demographics/av/{commune_id} .
select * from avellinodemographics;
```

MAPID-Commune-Demographics-ca

```
:Commune/ca/{commune_id} :hasDemographic :Demographics/ca/{commune_id} .
select * from casertademographics;
```

MAPID-Commune-Demographics-sa

```
:Commune/sa/{commune_id} :hasDemographic :Demographics/sa/{commune_id} .
select * from salernodemographics;
```

MAPID-Commune-Development-av

```
:Commune/av/{commune_id} :hasDevelopment :Development/av/{commune_id} .
select * from avellinodevelopment;
```

MAPID-Commune-Development-ca

```
:Commune/ca/{commune_id} :hasDevelopment :Development/ca/{commune_id} .
select * from casertadevelopment;
```

MAPID-Commune-Development-sa

```
:Commune/sa/{commune_id} :hasDevelopment :Development/sa/{commune_id} .
select * from salernodevelopment;
```

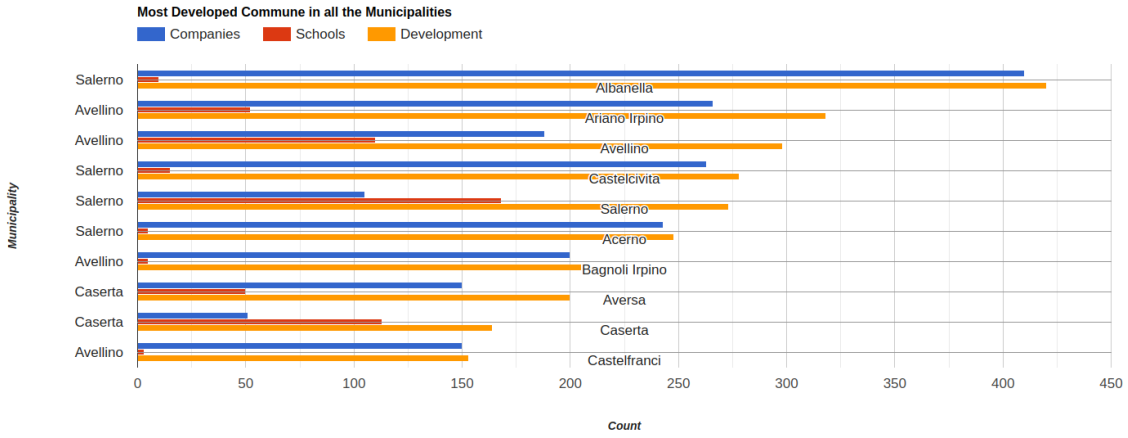
SPARQL queries

We answer the questions about the domain using sparql queries.

Q1: Which commune is the most developed in terms of schools and companies?
Answer: Top 10 commune in terms of schools and companies

Query

```
PREFIX : <http://www.semanticweb.org/ontologies/2022/5/DI#>
SELECT ?municipality_name ?commune_name (?number_of_companies AS ?Companies) (?number_of_schools AS ?Schools) (?number_of_companies + ?number_of_schools as ?Development)
{
    ?m a :Municipality.
    ?m :nome ?municipality_name.
    ?m :hasCommune ?c.
    ?c :nome ?commune_name.
    ?c :hasDevelopment ?d.
    ?d :numero_di_aziende ?number_of_companies.
    ?d :numero_di_scuole ?number_of_schools.
}

ORDER BY DESC(?Development)

LIMIT 10

```



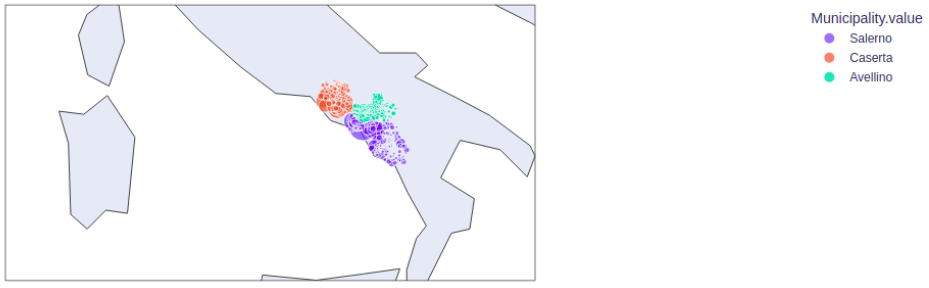
Q2: Which is the most populous commune presented on the map?

Query

```
PREFIX : <http://www.semanticweb.org/ontologies/2022/5/DI#>
SELECT (?municipality_name AS ?Municipality) (?commune_name AS ?Commune) ?lat ?long (?number_of_inhabitants AS ?Population)
{
    ?m a :Municipality.
    ?m :nome ?municipality_name.
    ?m :hasCommune ?c.
    ?c :nome ?commune_name.
    ?c :hasDemographic ?d.
    ?d :numero_di_abitanti ?number_of_inhabitants.
    ?c :hasGeography ?g.
    ?g :latitude ?lat.
    ?g :longitude ?long


}

ORDER BY DESC(?Population)

LIMIT 10

```

	Municipality	Commune	lat	long	Population
1	Salerno	Salerno	"40.682419254128725"^^xsd:double	"14.75935319328574"^^xsd:double	"130555"^^xsd:integer
2	Caserta	Caserta	"41.071956309748"^^xsd:double	"14.331304375950188"^^xsd:double	"73398"^^xsd:integer
3	Avellino	Avellino	"40.91800963711942"^^xsd:double	"14.786731772710974"^^xsd:double	"54561"^^xsd:integer
4	Caserta	Aversa	"40.97314297157902"^^xsd:double	"14.207191321232848"^^xsd:double	"51228"^^xsd:integer
5	Salerno	Cava de' Tirreni	"40.757885655784186"^^xsd:double	"14.710207674674953"^^xsd:double	"50840"^^xsd:integer
6	Salerno	Battipaglia	"40.630591177280365"^^xsd:double	"14.976096287338477"^^xsd:double	"50780"^^xsd:integer
7	Salerno	Scafati	"40.77008047562105"^^xsd:double	"14.53251623328741"^^xsd:double	"49236"^^xsd:integer
8	Salerno	Nocera Inferiore	"40.74614524841309"^^xsd:double	"14.635658741608497"^^xsd:double	"45952"^^xsd:integer
9	Caserta	Marcianise	"41.03613609151088"^^xsd:double	"14.299489299999998"^^xsd:double	"39386"^^xsd:integer
10	Salerno	Eboli	"40.615396"^^xsd:double	"15.054934"^^xsd:double	"38525"^^xsd:integer

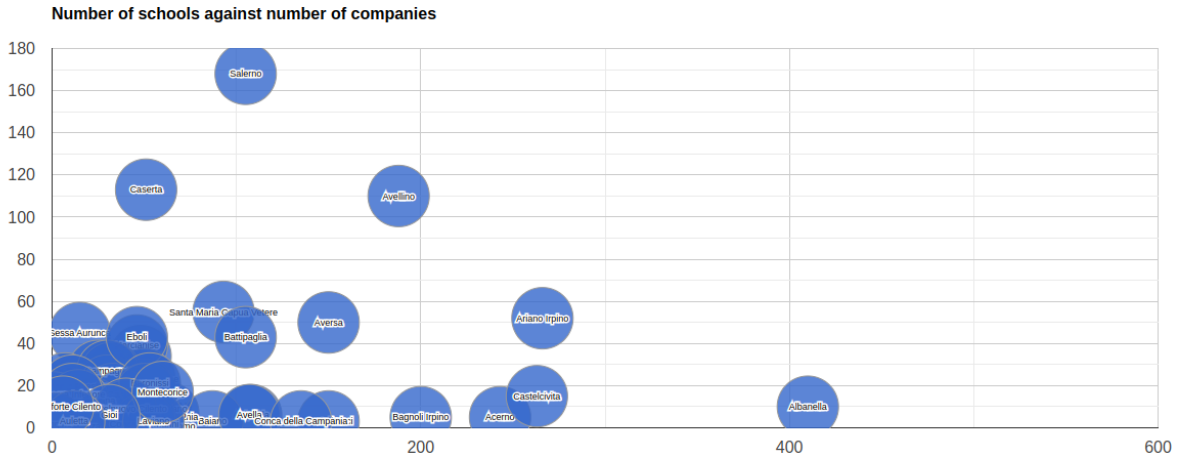
Population of different Commune in all the given Municipalities



Q3: How are the number of schools and the number of companies related?

Query x +

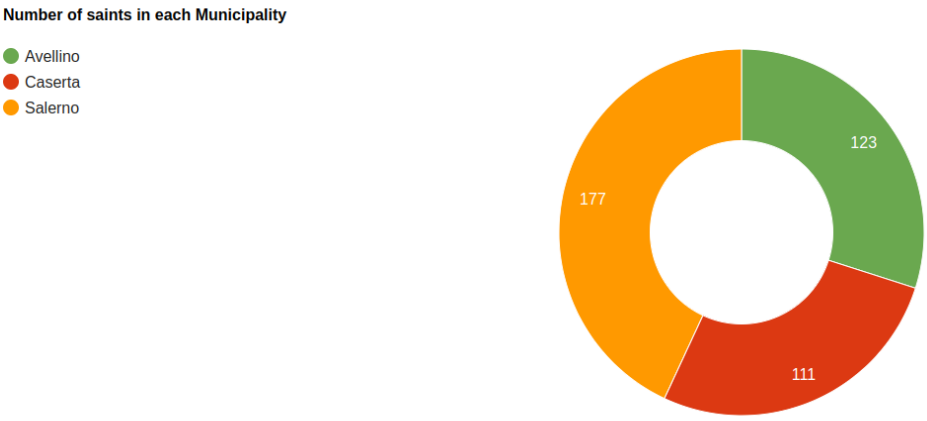
```
PREFIX : <http://www.semanticweb.org/ontologies/2022/5/DI#>

SELECT ?commune_name (?number_of_companies AS ?Companies) (?number_of_schools AS ?Schools)
{
    ?m a :Municipality.
    ?m :nome ?municipality_name.
    ?m :hasCommune ?c.
    ?c :nome ?commune_name.
    ?c :hasDevelopment ?d.
    ?d :numero_di_aziende ?number_of_companies.
    ?d :numero_di_scuole ?number_of_schools.

}
```



Q4: Which region has the least number of saints?

Query x +

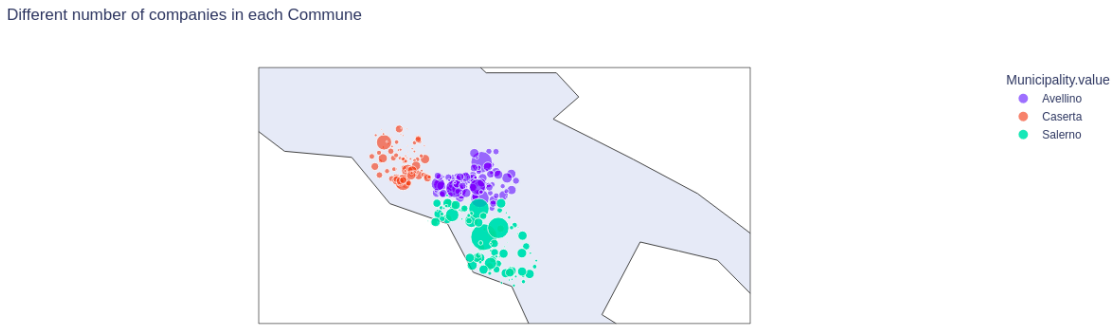
```
PREFIX : <http://www.semanticweb.org/ontologies/2022/5/DI#>
SELECT ?nome (Count(?m) AS ?saint_count)
{
    ?m a :Municipality.
    ?m :nome ?nome.
    ?m :hasCommune ?c.
    ?c :hasSaint ?s.
}
GROUP BY ?nome
```



Q5: Do latitude and Longitude have a relation to the number of companies?

Query

```
PREFIX : <http://www.semanticweb.org/ontologies/2022/5/DI#>
SELECT (?municipality_name AS ?Municipality) (?commune_name AS ?Commune) ?lat ?long (?number_of_companies AS ?Companies)
{
    ?m a :Municipality.
    ?m :nome ?municipality_name.
    ?m :hasCommune ?c.
    ?c :nome ?commune_name.
    ?c :hasDevelopment ?d.
    ?d :numero_di_aziende ?number_of_companies.
    ?c :hasGeography ?g.
    ?g :latitude ?lat.
    ?g :longitude ?long.


}
```



Web application

A simple python web application was developed for displaying visualizations and facts about the domain, It makes use of Ontop as a SPARQL endpoint to query the database through the ontology, extracting relevant integrated information and allowing some

interactive visualizations and presenting facts about the domain.

Ontop system is connected to the datasource (present in the form of a database) in PostgreSQL. The mappings written in Ontop are then validated. The data for the application is made available via SPARQL queries written over the datasource.

The web application is built using Streamlit (an open-source app framework in Python language. It helps us create web apps for data science).

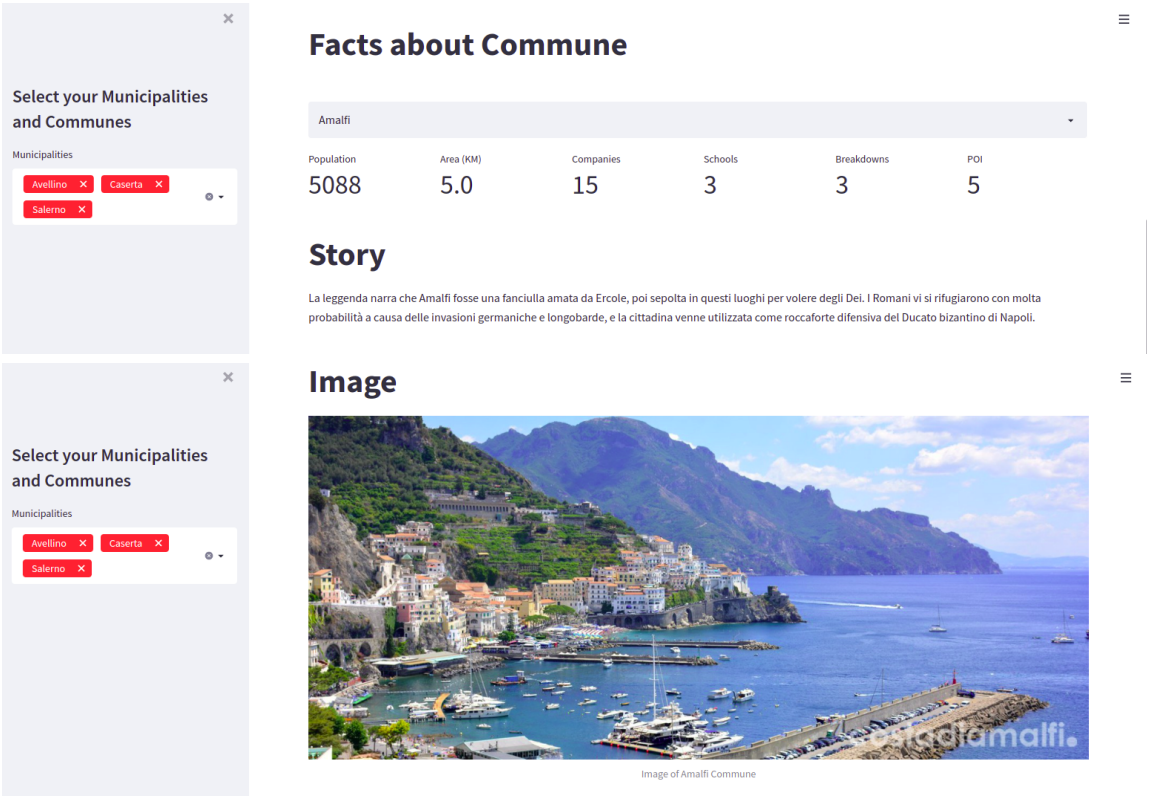
Some examples of the visualizations:

We provide a dropdown to select the option of choosing all the municipalities on the left.

Under statistics, we provide map visualization based on i) Population ii) Area iii) Companies
iv) schools v) Breakdowns vi)POI (points of interest)

We visualize the companies present in all the communes in all the municipalities and compare among them by placing them on the map

We display facts, a brief story and image about the selected Amalfi commune



References

[1] Jean-Baptiste, Lamy. "Ontologies with python." *Apress, Berkeley, CA* (2021)
[2] https://ontop-vkg.org/guide/getting-started.html
[3] Lectures slides