UNIVERSITY OF PADOVA

Department of Mathematics

Master Thesis in Data Science

Detection and Analysis of Unused Properties: Real Estate Analytics for the city of Padova

Supervisor
Professor Alberto Roverato

Master Candidate
Camilla Colanero

Academic Year
2023/2024

Abstract

The issue of unused properties causes significant challenges in urban environments, affecting economic growth, social equity and sustainable development. This study focuses on the city of Padova, where, despite a high demand for housing, a considerable number of properties remain vacant or underutilized. The research aims to quantify the extent of this problem in the city and identify key factors contributing to property usage.
By working with a comprehensive database provided by the Municipality of Padova, including Population and Land Registry data, the study implements various data cleaning and preparation techniques to create a reliable and accurate dataset. A binary classification model is then employed to categorize properties as used or unused, with water consumption data utilized as the primary indicator of usage.
The findings of this study provide valuable insights for urban policy makers, highlighting the necessity of specific strategies to encourage the utilization of unused properties in order to address the housing issue in Padova. The methodologies and results presented in this dissertation aim to be used as a model for similar studies in other cities facing analogous challenges.

Contents

List of Figures

1 Introduction

The problem of unused properties presents a significant challenge in urban areas worldwide, impacting economic growth, social equity and sustainable development. Despite a high demand for housing, many properties remain vacant or underutilized. It has been found that in 2014, in Europe, about 11 million homes were estimated to be unoccupied, with an average above 20% in Mediterranean countries [1]. One of the causes of this phenomenon can be attributed to the declining birth rate and the consequent decrease in the population. In Italy in 2019 more than 10 millions of houses were estimated as unused [2].
The paradox of high demand while many unused properties is evident in Padova, a city known for its university that attracts thousands of students annually. Despite this, the city faces a persistent issue of unused properties, which contributes to a difficult housing market, inflated rental prices and many people that do not find accommodation.
The Municipality of Padova, in order to understand better the actual magnitude of the problem and then find some solutions to face it, has decided to undertake a study aiming to find how many unused houses there are in the city and which can be the variables that explain the usage of a property. The goal, after understanding the vastness of the problem, is then to implement more specific policies that encourage citizen to sell or rent their unused properties. Indeed, some strategies have already been implemented in the last years, as for examples some tax advantages to those who rent to students. However, the Municipality of Padova cares deeply about this issue and wants to improve even more its policies.
This dissertation has thus the scope of reporting the study conducted and

its findings, hoping that it will be of example for future researches in other cities.
We will use various sources of knowledge of the Municipality, such as for instance the Population Register and the Land Register. Since not all the sources of information are updated to the current date, the study refers to the year 2021, for which all data were available. The main focus of the analysis has been preparing the data, by using processes of data cleaning and data preprocessing in order to create clean and accurate tables that can be reused also for other scopes of the Municipality, and then performing a binary classification to categorize the properties in *used* and *unused*, in order to understand the magnitude of the problem.
Before delving into the study, we report the definition of unused property that we will use in order to correctly classify each house. Indeed, a not utilized house is not only a property that is never used or is abandoned. We will base our classification on the usage of the object, found through the amount of water consumed: a property in our study is classified as *unused* if the quantity of water utilized is below the minimum required for the basic needs of one resident. This means that we will set as not used also a property that is highly underutilized.
In the dissertation, we will first provide a description of the database that provides most of the information, with a more in depth illustration of the datasets used, then we will describe the data preparation techniques utilized in order to connect all the different sources of knowledge and, at last, we will report the models trained for the binary classification, with a description of the results found.

2
The database

In this chapter we will describe the database that will be the fundamental source of data for our study. For a more specific description of the tables used for the analysis, the reader can refer to Chapter 3.

The Municipality of Padova, for the upkeep and maintenance of data coming from different internal sources, is sustained by a service of the Engineering Group, a Digital Transformation Company leader in Italy; this service is called Municipia. Municipia supports cities of all sizes in the digital transformation process by creating innovative services through private investments and the absorption of operational risk [3]. The key scope of this part of the Engineering Group is indeed to support the development of cities into smart cities.
The database in question is a product of Municipia, especially tailored for the Municipality of Padova. The name of the database is ACSOR (that is the acronym for Anagrafe Comunale dei Soggetti, degli Oggetti e delle Relazioni) and it is the Municipal Registry of the Subjects, of the Objects and of the Relations.
The scope of this database is to arrive to a reconciliation of the information obtained from the data coming from both the Municipality and other supra-municipal sources like the Province, the Region or the Land Registry. Furthermore, there is a frequent periodical update of data in order to have always the newest information available.
The acquired data from the different sources (also referred to as environments) are originally registered in a staging area, before being elaborated in a reconciled area. Therefore in the staging area we have a copy of the

data as they have been obtained, while in the reconciled area we have the values after the process of data cleaning and variables selection. Data cleaning involves identifying and correcting errors or inconsistencies in the data, such as duplicate records, missing values, or incorrect entries. This step is essential for ensuring the integrity and quality of the data before it is used for analysis or decision-making. Variable selection, on the other hand, involves choosing the most relevant variables from the raw data for inclusion in the final dataset. This ensures that the data used in the final tables is of high quality and relevance. Furthermore, in order to create the final tables, the data are divided into three different structures: the subjects, the objects and the relations. Indeed, for every data source there is more than one table related to it. Moreover, for each table the variables belong to one of these structures, based on the characteristic that they supply. To understand better this concept we provide the following example.
For the Population Registry source we have:

- **Subject** : surname, name, Tax ID Code, date of birth, date of death, ...
- **Object** : home address, house number, floor number, ...
- **Relation** : start date, end date, family code, ...

Hence, the subjects include the information about natural persons and legal persons (either private or public organizations), the objects refer to buildings, residences and all kind of immovable properties and the relations are the legal bonds between a subject and an object (in the form of ownership or usage) or a subject and another subject (kinship).
The reconciliation of the various data sources, however, is not straightforward. In order to state that a subject (or an object) of an environment is the same of another source, some quality checks must take place. We report here the approach used for the subjects, that is similar to the one employed for the objects.
In numerous cases, due to errors or missing data, the comparison between subjects is very laborious. In order to state that two subjects coming from two different sources are indeed the same person some operations of record matching are employed. First, there is the choice of the variables that will be compared (for the subject we have for example the name, the surname, the Tax ID Code, the date of birth ...). For each record then the correspondence produces a score from 0 to 100, where 100 represents a certain comparison while 0 means that the two records are surely different. Then there is an

average of the scores, that produces the probability that the two subjects are the same one: the more records match, the higher is the probability. Once the score exceeds a certain threshold, the match is confirmed. There could also be some fixed condition that automatically reject certain associations. After these operations, the final tables are created.
As previously said, for all the data sources there is a copy of the acquired data in the database. Each one of these copies has a variable that operates as the identifier of the subject or the object that is specific of the data source. For example, the fictional subject Mario Rossi in the Population Registry environment could be referred to as the subject with identifier 123456 while in the TARI source (that is the source of the waste tax) he could have the ID number 200000. Thus, in order to identify more easily, and in a unique way, the subject (or object) in the final tables created after the reconciled area, a key identifier specific of the ACSOR environment is created. The new ID is named *IDR_SOG* for the subjects and *IDR_OGG* for the objects, while the identifiers of the input sources are called *IDR_SOG_STL* and *IDR_OGG_STL* respectively.
In the database, as we will see in the next chapter, we can also find some tables that link the identifiers from all the various sources to the *IDR_SOG* of the ACSOR (and the same thing applies for the objects). This tables are of primary importance to move from a data source to another. By maintaining a global mapping of the identifiers, the ACSOR database ensures the accessibility of data across different sources, facilitating data retrieval and analysis.
The ACSOR database is therefore the main source of data for our analysis. However, as we will explain later more in detail, the information that come from this database is not always complete and reliable, and moreover not all the data sources that are needed for our study can be found inside it. Indeed, we consider two more sources for the data regarding the Land Registry and the water bills. The former is actually present in the ACSOR database, however the data of the tables related to it cannot be considered reliable, for reasons we will discuss in the next chapter. Instead the information about the water bills is not reported in the ACSOR database, so we had to extract those data manually from another source.
For the Land Registry dataset we extracted the records from an internal software of the Municipality of Padova that allows the consultation of the Land Registry, called CARTO ACI. CARTO not only provides detailed historical data but also offers advanced features for data analysis and visualization, making it a valuable tool for urban planning and development. In the software CARTO ACI we can indeed find all the information about the objects

present in Padova through the different years, with attributes very similar to the one found in the tables of the ACSOR database. As a matter of fact, from this software we were able to download a dataset that contains all the objects that were present in Padova in 2021, and this table turns out to be much more reliable than the previous one considered.
On the other hand, data related to the water bills in the Municipality of Padova are not retained in the ACSOR database, hence we had to extract these records from SIATEL (acronym for *Sistema Interscambio Anagrafe Tributarie Enti Locali* that translated is Local Authority Tax Registry Exchange System), that is a telematic connection system of the Agenzia delle Entrate created by the Ministry of Economy and Finance which allows the active exchange of personal and tax information between central and local public administrations. The resulted data were in the form of fixed position text, that means that, for each row, from a position i to a position j we can find a specific information. In order to convert this form of data into a more readable table, we used a document outlining the correspondences of each sequence of positions to the information related (for instance from position 3 to position 6 we can find the year to which the bill refers to). After the transformation, we obtained a dataset with specific columns, as we will see in the next chapter.
However, the concern that comes from taking data from sources that are different from the ACSOR database is that they do not have the ACSOR identifiers *IDR_OGG* and *IDR_SOG*. Both the other sources have some identifiers of their own, used in order to distinguish their records, but for the purpose of continuing with our study we will need to find the ACSOR identifiers. This is due to the fact that, in order to relate an object to a subject, their ACSOR IDs are needed since the table that put them in relation is based on those.
This is not an insurmountable problem, but it is a delicate one because the only information on which the association can be based on is in the form of a string, that is not always written according to consistent rules. For instance, variations in spelling, position of words in the phrase, or data entry errors can complicate the matching process. However we will discuss more in depth this problem and its resolution in the next chapters.

3 The datasets

In this chapter we will describe in details how the datasets that we have used for our study are structured. A list of all the important variables of each table with a brief description of the content can be found in the Appendix.

1 Land registry

The Land Registry (or Cadastre) is a publicly available record where ownership and other rights related to immovable objects are stored [4]. In the land registry both lands and buildings are listed but for our scope we will consider only the objects registered as buildings. In particular, the land registry of the Municipality of Padova is a collection of all the stages of all the objects of the city of Padova. The stage of an object is defined as the characteristics of an object before a variation; for each change in one of the characteristics we have a new stage. The variations can be manifold, for instance we could have a topographic alteration, where a new name of the street is given, or a structural modification, after a renovation.
In the ACSOR database, we have different tables that refer to various aspects of the land registry, that is too complex to be put in one single table. All of these tables can be joined using the unique key *IDR_OGG*, that is the distinctive ID of the object in the database ACSOR.
The main table of the land registry in ACSOR is OCAT (that is an acronym for the italian translation of Cadastral objects), a dataset in which all the objects and their stages are listed with the dates of validity of the latter. However, after analyzing the dataset, we found out multiple problems, such as, for example, many rows in which the start date of validity was higher

or equal than the end date. Moreover, the core issue that occurred was that some of the objects (after a join with the dataset INDO, that we will describe shortly) were referring to addresses outside of the municipality of Padova (e.g. we found some objects that are located in Cittadella or Piove di Sacco), even though they were classified with the code G224 that is the identifier of the city of Padova. Indeed, the number of objects that in theory should have been in Padova in 2021 was too big to practically make sense (we had around 700000 rows).
For these reasons, we decided to use another dataset, that does not come from the ACSOR database but from a software used in the municipality offices that is called CARTO ACI, as we already mentioned in the previous chapter. This table is very similar to the one of the ACSOR, but the values are more accurate. As a matter of fact, the number of objects in this dataset is much more reasonable.
The only inconvenience is that, since the dataset does not come from the ACSOR database, there is not the usual identifier that makes the join of the table easier. Therefore, we will need to find a way to associate the objects of the CARTO dataset to theirs *IDR_OGG*. The process is explained in the next chapter.
The following is a description of the tables, that refer to the land registry, that we will use.

1.1 IDCT

This ACSOR table is the dataset in which the variables related to the land survey plan are stated (the acronym stands for *IDentificatori CaTasto*, that is *identifiers for the land registry*). The land survey plan is a document that legally determines the boundaries of properties and assigns to each object some key values in order to identify it spatially. These values have specific names that cannot be translated in other languages, therefore we will report the original names in Italian. What is important to keep in mind is that the combination of these values refers to only one object: if for example we have a flat complex, the combination of the four variables that we will list will be different for each distinct flat.

1.2 UIU

Another useful table of the ACSOR database for our aim is the UIU dataset (where UIU stands for *Unità immobiliari urbane*, that is *urban real estate units*), in which the information about the characteristics of the objects

is stored. In this table we can, as a matter of fact, find the type of the object. There are five categories into which immovable properties are classified: A (private houses), B (buildings for collective use, such as schools and barracks), C (commercial buildings, such as garages, shops, shelters), D (industrial buildings), E (special buildings). In each category there are various sub-categories. The objects that we will analyze belong to the first category.

1.3 INDO

In this ACSOR table the objects are associated to an address. This dataset is fundamental because it will allow us to distinguish among objects belonging to a subject when we will perform some joins between tables.
However, the way in which the address is written is not always the same among datasets. For example, in one table we could find the name *Via Augusto Anfossi* while in another one *Via A. Anfossi*, therefore we cannot make a punctual comparison among the fields that pertain the name of the addresses, but we will need to come up with some other ways to find if the names refer indeed to the same address.
To each object there could be associated more than one name of a street, however there is only one principal address for each one (the others are categorized as secondary).

1.4 MW_RICERCA_FUSIONI

During the years, some of the IDs of the objects, *IDR_OGG*, have been merged with other identifiers (we can think for example of two distinct apartments that, after a renovation, become one). In our analysis, in order to not consider some duplicates, we need to be aware of these merges. Indeed, in this dataset we find this information.

1.5 CARTO

As before mentioned, this dataset is taken from the software CARTO ACI and it is a collection of all the objects that were present in Padova in 2021. This table serves as the base for all our future analysis; as a matter of fact, since our objective is to find out if an object in the municipality of Padova is used or unused, we need an initial dataset to classify.
Analogously to the IDCT dataset, the objects of CARTO are distinguished by the same four spatial identifiers. Actually, most of the rows of this dataset do not have the value for the first spatial identifier, the *sezione* variable. This

will make the association of the objects to their *IDR_OGG* somewhat more challenging, but the other three identifiers and the address are enough to make a correspondence.

2 Population registry

Another important dataset for our goal is the Population Registry dataset (*Anagrafe* in Italian), where all the demographic information about a person is, as the name recall, registered. The term "population register" was defined in 1969 as *"an individualized data system, that is, a mechanism of continuous recording, and/or of coordinated linkage, of selected information pertaining to each member of the resident population of a country in such a way to provide the possibility of determining up-to-date information concerning the size and characteristics of that population at selected time intervals"* [5].
In particular in our case we have the information about all the people that, sooner or later, have been registered in the city od Padova. This means that either the person was born in Padova or has at least legally resided there for a brief amount of time.
In the registry, all the variations of information of a subject are retained: for example, if a person in a certain date moves in another house and therefore changes the residential address, a new row will be created with the new updated details.
We have two different datasets that refer to the registry: the first one is called ANPO (that stands for *ANagrafe POpolazione*, or *Population Registry*) and inside it there is only one row for each subject, the one with the most recent information. In the second one, ANPOS (the *hystorical Population Registry*), we can also find all the historical details. We will need both of this datasets because ANPO is updated to the current day while we need to know who was living in Padova in 2021, therefore if a person has moved after 2021, the information that we need is kept in ANPOS.
The structure of both this datasets is the same, except for one further column in ANPOS where a sequential number is kept.
Another table that is related to the Population Registry but that provides different information from the ones of ANPO is the dataset SOGG. In this table we cannot find the address of residency of the subjects but we can find a description of both physical and legal persons; in particular we will need the information about the VAT code.

3 Relations

The ownership or use of an object from a subject is retained in some specific tables of the ACSOR database. An object is in relation with a subject if there is a legal form of ownership or usage. The main purpose of these tables is to connect the *IDR_OGG* to the *IDR_SOG*, and also to tie the identifiers specific of the other environments (e.g. the Population Register) to the identifiers of the ACSOR domain.
The tables that accomplish this task are the following:

3.1 RESS

This table allows to connect the ACSOR identifier of the subject *IDR_SOG* to the ones of the other domains *IDR_SOG_STL*; in particular we will need the association with the *ANAPOP* source, that is the one of the Population Registry, because, as already mentioned, inside the ANPO an ANPOS datasets it is the only identifier that we can find.

3.2 RUP

In this table we can find the relations between objects and subjects. This is one of the most important tables since it allows us to understand which objects are associated to a subject, and therefore it enable us to join the datasets that include only the subjects to the ones where we have only the objects.
In this dataset, for some of the *IDR_OGG* not all the different data sources are present. For this reason we will keep all the relations between a subject and an object, even if they are from a data source that is not related to our analysis, since it means that at least in some way that object is linked to that subject.

4 Water bills

Italy's water supply is centralized and public, with a smaller portion of private companies. Drinking water has no cost per se, it is an essential good, however there are numerous services that enable the public use. These services' costs are the reason why there is a water bill.
The water bill must be paid for all the immovable objects that receive a water supply, therefore this dataset is very important for our scope. Indeed

it allows us to understand in which houses there is an expense of utilities, and this points to the state of use of the object.
The data from the water bills in the Municipality of Padova are not retained in the ACSOR database, hence we had to extract the records from SIATEL, as previousy mentioned in Chapter 2.
Similar to the CARTO dataset, since the water bills data are not in the ACSOR environment, we do not have directly the identifiers for both the objects and the subjects, but we need to find them trough associations.

4
Data preparation

The main workload of this study has entailed the data cleaning and data preparation processes, before we could work on the models for our classification. In this chapter we will present the transformations performed in order to arrive to the final dataset on which the classification could take place.
As mentioned in the previous chapter, we will consider and join multiple tables, but the main fields around which our data stand are three: the list of the objects, the dataset of the population and the information about the water bills. For all these fields we will create a specific table, and these dataset will interact between each other. However, the most important dataset, that will be the base on which we will perform the classification, is the list of the objects.
In order to have user-friendly and understandable data, many challenges have arisen. In this chapter we will describe them and how we have been able to face them. We will divide the chapter into the three aforementioned categories, delving deep into the issues presented for each one, and then we will show the final dataset on which we will perform the classification. The models will be displayed in the next chapter.
From now on we will mention different datasets and their variables; in order to follow along, the reader can refer to Chapter 3 and the Appendix.
Before examining deeper the preparation of the datasets, we present the software tools that supported the development of our dissertation. The following is a description of the key tools and their roles in our study:

Jupyter Notebook

Jupyter Notebook is a web-based interactive computing platform. This tool has been the main environment used for coding, data analysis and data visualization. In particular, Python has been the programming language used throughout all the study. We have utilized many different libraries, but the following were especially important:

- *pandas*: it is a library especially tailored for data manipulation and analysis. It has been the principal element used for data cleaning and data preparation (Chapter 4).
- *scikit-learn*: it is a free and open-source machine learning library that we used for the modelling part of the dissertation (Chapter 5).
- *matplotlib* and *seaborn*: they are both Python data visualization libraries, that we used to construct the plots throughout our analysis.

Jupyter Notebook thus has provided a comprehensive environment that has supported all phases of the analysis, from data preparation to model evaluation and visualization.

DBeaver

DBeaver is a universal database management tool and it is the SQL client used to interact with the databases. This software has been used mainly because it provides a user-friendly interface for managing SQL queries, exploring database schemas and visualizing query results. It has been fundamental in order to navigate among the tables of the ACSOR database. Indeed, the core functionalities that have been very useful for our study are:

- Database management: navigating through the database structures, understanding the relationships between tables and performing specific queries.
- Query development: writing and testing SQL queries in a more interactive and visual environment before integrating them into the Python code.

DBeaver's functionalities made the data preparation process more efficient and accessible, in particular during the first part of data preparation and joining of the tables.

ArcGIS Pro

ArcGIS Pro is a professional desktop GIS application from Esri, where a GIS (Geographic information system) software is a program that provides the ability to create, analyze and visualize geographic data. We have used this software for visualizing spatial patterns, particularly when examining property usage across different neighborhoods. Furthermore, it has been useful for finding the neighborhoods in which the objects are situated. The main operations have entailed:

- Geographic data handling: manipulating and analyzing spatial data such as property locations and neighborhood boundaries.
- Map creation: creating detailed maps that visualized the distribution of used and unused properties, providing geographic context to our results.

ArcGIS has given us the tools needed to include spatial data in the analysis. This improved the understanding of our results and of geographic patterns in property usage.

1 List of objects

The main goal for this part is to create the list of all the immovable properties, classified as private houses, that were present in the municipality of Padova in 2021, along with their *IDR_OGG*. As already seen in Chapter 3, the dataset CARTO (also referred to as C from now on) already has a list of all the objects referred to the city and year in question, but without the ACSOR identifier. The main challenge for this category is therefore associating the CARTO object to the ACSOR one.
First of all we performed a filtration of the CARTO dataset, in which we kept only the records related to private houses (i.e where the *categoria* value is one of the following: A1, A2, A3, A4, A5, A6, A7, A8 or A9). The total number of rows after this filter is around 150000.
The first step with the purpose of creating the association with the ACSOR objects was to create the dataset MAPPALE (from now on referred to also as M), by joining the two tables INDO and IDCT on the primary key *IDR_OGG*. The next stage was to join the CARTO dataset with the MAPPALE one. This join has been performed on the variables *C.foglio, C.mappale* and *C.subalterno* with *M.FOG, M.MAP* and *M.SUB* respectively. As seen in the previous chapter, these sets of variables (along with

the *sezione*) are unique spatial identifiers of an object. However, since the *sezione* variable is almost never present in the CARTO dataset, we could not make a unique association using all four identifiers. We hence had to join the two datasets only on three out of the four IDs, but in this way one object of the C table could have been associated to more ACSOR objects. For instance, if we have a record in the CARTO dataset with *foglio* 1, *mappale* 67 and *subalterno* 34 and without the value of *sezione*, and in the MAPPALE table we find two distinct objects (with different *IDR_OGG*) with the same spatial identifiers but one with *sezione* 5 and the other with *sezione* 1, both these objects will be associated to the same record. Moreover, two records in the C dataset can have a missing value for the *sezione* and the other three spatial identifiers equal.
What we have just presented is the core issue with the list of objects. In order to correctly associate an object of the CARTO dataset to its ACSOR one, we will have then to compare also the home address, both with the name and the number. For this reason, we created two new columns only for the home number of the *indirizzo_catasto* and *indirizzo_comune* variables, so that we could perform a match also on that, paying particular attention to the addresses in which a number is already part of the denomination (e.g. the address *Via 20 aprile 1944 13*, where the home number is 13 and not 1944 neither 20) and also to keeping only the number (for instance if there is an address *Via Anfossi 15/A* we only need the number 15).
Going back to the joining of the tables M and C, we had to transform the type of the spatial identifiers in order to make them comparable in the two datasets. Indeed, in the two tables those variables were written with different approaches, for instance a string with many blank spaces at the end in one while a number in the other, so we performed some data cleaning to make them equal. After the join, the resulted number of rows had nearly quadrupled, and this is due to what we have already mentioned about the non-uniqueness of the associations. We then selected only the columns of interest (the ones described in the Appendix) and we dropped the duplicates, arriving to a table with around 400000 rows.
Before comparing the objects through the addresses, we used the information provided by the table MW_RICERCA_FUSIONI, about the objects that have been merged. Indeed, an *IDR_OGG* that is no longer existing will have the same positional identifiers of the object to which it has been merged to. We hence joined our table with the dataset of the merges; then we removed the records in which there was an object (let's call it A) that had been merged to another object (let's call it B) if the object B was already present in the dataset. Instead, if no other record of B was present, we changed the

value A with the value B.
In our dataset, however, there were still some duplicates where the only different variable was the number of the stage of the object *STA*. We thus decided to keep only one of the duplicated rows, the one with the highest stage. After this process, the number of records of the dataset dropped to approximately 200000.
The next step was to create a new column, called *status*, in which we reported the status of the association. This column is made of categorical variables that can have the following values:

- ***no match*** if the object of the CARTO dataset has not been mapped to any *IDR_OGG* or, if it has been associated, the name of the address *DEN_VIA* of the ACSOR object is a null value (thus we cannot use it to make a comparison).
- ***correct match*** if the address of the CARTO object correspond to the one of the ACSOR in both the name and the number.
- ***address to check*** if the home number of the ACSOR object is the same as at least one between those of the *indirizzo_catasto* and *indirizzo_comune* but the name of the address is different.
- ***wrong number*** if the house number does not match. In this category fall both the addresses in which the name corresponds and the ones were the name is totally different.

In order to set the status, we performed a comparison among the different addresses *DEN_VIA*, *indirizzo_catasto* and *indirizzo_comune*. The main challenge has been understanding if the different ways in which the name of the address was written were referring to the same one. Indeed, we could not perform a direct comparison between the addresses, but we had to come up with a different strategy. Since in most cases the values of the *DEN_VIA* variable were the abbreviated name of the address (e.g *Via A. Anfossi* instead of *Via Augusto Anfossi*), the principal criteria that we have decided to adopt in order to compare the names was to confront the set of the first letter of each word, without considering the order. In this way, even the addresses that are equal but written in a different order (for instance *Via Jacopo Facciolati* in one case and *Via Facciolati Jacopo* in the other) were identified as exact.
Nevertheless, in this way not all the right matches were correctly identified. Indeed, if the names of the addresses differed for some words, perhaps because some parts of the address were missing (for instance *Via P. Ponchia*

instead of *Via Monsignor Placido Ponchia*), or were written in different ways (e.g *Via Iesolo* in place of *Via Jesolo*), the previous criteria would not identify this cases as correct, while they actually are. For these reasons, we had to manually identify some matches as correct.
The next step was to eliminate the rows in which the correspondence was incorrect if there was present a record in which the spatial identifiers and the address were the same and, moreover, the status was *correct match*. In the following table we report an example of a situation that we could meet:

FOG	*MAP*	*SUB*	*indirizzo_comune*	*DEN_VIA*	*IDR_OGG*	*status*	
1	37	3	Via Guizza 7	Via Guizza	234278	correct match	✓
1	37	3	Via Guizza 7	Via Arca	128345	address to check	×
1	37	3	Via Guizza 7	Via Guasti	82399	wrong number	×

Indeed, if for a CARTO object, that has some specific spatial identifiers and a certain name of the address, we have found at least one ACSOR *IDR_OGG* that is a correct match, we can discard all the other records with the same spatial IDs and home name that are not right. After these eliminations we arrive to a dataset with around 153000 rows.
The last step has been to add to each record the surface area of the object from the table UIU. This variable will be important in our future study.
We then performed an ulterior inquiry, *a posteriori*, in order to check if we missed some information that was present in the CARTO dataset. We found out that we were not able to assign an *IDR_OGG* only to 29 rows of the initial CARTO dataset, therefore this result is quite acceptable.
However, even in the initial CARTO dataset there were some repetitive records, in which the values of the variables were pretty much the same but the name of the address was somehow different (e.g. we have a record with the *indirizzo_catasto* variable set to *Via Candiano II Pietro* and another with *Via Pietro Candiano Secondo*). We were keeping these repeated records only in order to do the *a posteriori* check, however these rows are redundant so we performed some data cleaning processes and we arrived to a dataset of about 150000 rows.
As we will see shortly, this does not mean that we have 150000 distinct objects (if we investigate how many different *IDR_OGG* are present we find indeed a lower number of about 140000) because the same *IDR_OGG* can be associated to more than one address, as we have seen in the description of the table INDO in Chapter 3. However we will need the different denominations of the addresses in order to perform a better match with the other datasets.

We will keep only one row for each *IDR_OGG* at a later time.

2 Population Registry dataset

A key information for our analysis is the knowledge about the people that live inside an object. A priori we could infer that if an object is set to be the place of residence of somebody, it should not be unused. However, we will see if this inference is true only throughout our study.
The key goal for this part is thus to associate the object of residence to each subject of the Population Registry. In order to do so we will need an accurate list of all the people that were living in the municipality of Padova in 2021. As we have already mentioned in the previous chapter, we will use the tables ANPO and ANPOS. Indeed we filtered these two dataset by keeping the rows in which the variables *DAT_INI_RES_VIA* and *DAT_FIN_RES_VIA* included the year 2021. Since there were some errors that we wanted to control, we also included in our filter the rows in which the two aforementioned variables were equal or the end date was smaller than the start date of residency.
In total, from the ANPO table we filtered out about 229000 rows while from ANPOS we kept around 82000 records.
The first further filter that we performed was to remove the subjects for whom the date of death is before the first of January 2021, since they will not be of interest for our study.
One of the challenges that we had to deal with was the presence of errors in the ANPOS dataset. Indeed in many rows the end date of residency was either *99991231* or *00000000*. However, in the ANPOS table we can find the historical information, therefore the presence of a record in which the residency is not terminated is a signal of something irregular. Indeed, we would expect that if a subject is still living in a specific object, this information would be the most recent one and therefore it would be present in the ANPO table. Nevertheless, we could have this situation for different reasons, like:

1. Some information about the subject were modified (e.g. someone decided to change name), therefore in ANPO we find the same row with the new information.

2. Some previous mistakes, that do not include the date of residency, were changed (for instance the name of the address was spelled wrong), so in ANPO we find the correct information.

3. The date of residency, previously mistaken, was corrected. Here we can have two cases:

   (a) in ANPOS we find the wrong end date of residency and in ANPO the correct one is smaller than *20210101*.

   (b) in ANPOS we find the wrong end date of residency and in ANPO the accurate date is after the first of January 2021.

In the first two cases, we already have the record with the newest information possible in the filtered rows of ANPO, thus we can safely eliminate the "duplicated" ones from the ANPOS dataset. For the third reason, instead, it depends on the case: in case (a) the record is not present in the filtered rows of ANPO, nor it should be since the residency does not include the year 2021. Therefore the presence of the subject for our analysis is incorrect, thus we should eliminate the record. In case (b), on the contrary, the information is already present in our filtered ANPO, consequently also these rows should not be retained. Hence, as we have seen, no record that has a *DAT_FIN_RES_VIA* date undefined in the table ANPOS should be considered for our study.
For the other subjects of ANPOS with a determined end of residency date that have a record also in ANPO, something similar to what we presented above could happen. Indeed, there could be two cases:

1. The person has changed its residency in 2021, therefore the same subject will be associated to more than one object for the year in question.

2. There were some mistakes with the residency dates.

For the first case, we will keep the records coming from both ANPO and ANPOS since we are interested in knowing which houses were marked as residencies in 2021. In the second case, instead, we will keep only the information coming from the ANPO dataset: as a matter of fact, in this dataset the knowledge should be the most correct, thus if for a subject we have the same start date of residency in both datasets, but different end date, we will consider as the most accurate the one coming from ANPO.
Another error present in the historical dataset is that a few rows had the value of *DAT_INI_RES_VIA* set to 0. Throughout our analysis we found out that some of them were already present in ANPO with a corrected date of start residency and the same end date, thus we removed the rows in question from ANPOS. The remaining ones (only 3) were not present in ANPO and

had a end date greater than *20210101*, therefore we decided to keep these records.
A further challenge that we had to face was the fact that we could have more than one record in ANPOS related to the same subject. This could happen for the same two reasons presented above. However, if we have more than one row with the same subject and the same start of residency, we should retain only the record with the most accurate information. Thus we kept only the row that had the highest *DAT_VAR* value, that means that it was the most recent record amongst all.
As previously mentioned, in our initial filter we kept also the rows in which the date of end of residency was less or equal than the start date, because we wanted to check if these records would provide us with more knowledge. In case of equality, the dates of this rows could refer also to years that are different from 2021. In this situation we have three distinct cases that are worthy of our attention:

1. There is not any record in ANPO that is related to the same subject of the row in question. This means that probably at least one of the two variables has been changed and the period does not include 2021.

2. There is a more accurate record about the same subject in ANPO.

3. There is another record in ANPOS about the same subject, even if with a different name of the address, that has a start date bigger than the one of the row in question. This means that we know that the subject has lived somewhere else after the address with the wrong dates.

In all these cases we should remove the row from ANPOS, since it is a mistaken one. A similar approach can be carried out for the cases in which the start date is greater than the end date.
In the table ANPO there are some errors as well:

- There are some cases in which the values of *DAT_INI_RES_VIA* and *DAT_FIN_RES_VIA* are 0, yet also the name of the address is made of empty spaces, thus we will discard these rows since they will not be useful for our study.

- There are some rows with the start date greater or equal than the end date: we decided to remove the rows with mistakes that also had a null address and the ones in which the subject was born before 1916 even if the date of death was not registered.

The next step was to concatenate the remaining rows of the two tables in order to create the final list of residents. The ultimate number of rows before the aggregation was around 220000 for ANPO and 30000 for ANPOS. As previously mentioned, there could be more than one row related to the same subject, since a person could have changed home in the year 2021. Moreover, after an analysis of the variable *DEN_VIA*, we discovered that some of the addresses were not referring to the municipality of Padova (e.g. we have some addresses that start with *Rue*). This is due to the fact that in the ANPO dataset we have the most recent information about a subject, and therefore if the person emigrated abroad we have the record about the last address that is indeed foreign. This issue does not really affect our study since what we aim to do is to see if the objects of our list are marked as residencies, and not the opposite (i.e. find every object in which a subject has lived). Even with these 'foreign rows', we have that the total number of subjects coming from our list is around 240000, that is quite a reasonable quantity for the residents in 2021. The last step has been associating to the *IDR_SOG_STL* of each subject its ACSOR identifier *IDR_SOG*, using the RESS table.
The next target is to find out if, for each object of the list, we have at least one subject that resides inside it. In order to do this, we need to connect the list of objects to the dataset extracted from the Population Registry table. However, the association is not straightforward since we have no key variable to join the two datasets. Indeed, we need a third table that creates a bond between them: using the table RUP, we associated each subject to its related *IDR_OGG*, creating thus the key we were looking for.
With this process, we come across a challenge similar to the one encountered with the list of objects: for each subject we could have more than one object associated to it, thus in order to find the one that is listed as its residency, we have to compare the addresses. The criteria used for the comparison are similar to the ones described above in Section 1. Something worth of notice is that we compared the *DEN_VIA* of the Population Registry to both the *indirizzo_catasto* and *DEN_VIA* of the list of objects, especially in the cases in which the *status* value was different from *correct match*. This was done in order to have the most information possible available. Indeed, if in these cases we found a correspondence with the *indirizzo_catasto*, then we can infer that the *IDR_OGG* was correct and the mistake was only in the name of the street made by the ACSOR. On the contrary, if the *DEN_VIA* of the object was the right one, it means that the identifier is correct and the name of the address provided from CARTO was the wrong one.
We created a new column called *FLG_COR*, that takes values *True* if the

correspondence of the addresses is correct and *False* otherwise. The right compatibility is not univocal: indeed a subject can be linked to more than one object because of moving, and an object can be related to more than one subject because, for instance, a family lives in the same house.
The last step was to create some new variables that give the information about how many subject live inside the object. In order to have the greatest knowledge available, we decided to keep not only the information about the number of residents but also some characteristics about them. We will see in next chapter if these variables will be important for our study.
We created 6 new columns in which we divided the residents by age and sex traits:

- *under 19*: the number of subjects that live inside the object and that in 2021 were 18 years old or younger.
- *19-30 anni*: the number of subjects that live inside the object and that in 2021 were between 19 and 30 years old.
- *31-60 anni*: the number of subjects that live inside the object and that in 2021 were between 31 and 60 years old.
- *over 60*: the number of subjects that live inside the object and that in 2021 were older than 60.
- F: the number of females that live inside the object.
- M: the number of males that live inside the object.
- *residenti_totali*: the total number of subjects that live inside the object.

For the objects that had none or none correct association with a subject, all these variables were set to *0*. We then added these columns to our list of objects, because that is the dataset on which we will perform the classification.

3 Water bills dataset

Through the analysis and cleaning of this dataset we will be able to set the labels of *used* and *unused* in the list of objects. Indeed, from the levels of consumption of water we can assume the state of usage of a property. According to the World Health Organization (WHO), between 50 and 100

litres of water per person per day are needed to ensure that most basic needs are met and few health concerns arise [6]. We will use this datum as the threshold to understand if a property is used or not.
The main task for this part is thus to associate the water bills to the object to which they refer. Similarly to what we have seen in Section 1, since the water bills dataset does not come from the ACSOR database, the first course of action is to find the *IDR_OGG* of the object which the bill refers to. However, in this case it is even less straightforward since there are not even the spatial identifiers that we instead had for the CARTO dataset. Apparently, the only way to associate the bills to the objects seems to be by connecting the names of the addresses, but in this way we would not find the exact object (e.g. in one address we could find a condominium and there would not be any criteria to distinguish among flats). For an accurate connection, we should instead use the information about the subject that is the utility holder. In the variable *cf_tit_utenza* we can fin the Tax ID Code or the VAT number of the holder; indeed not all subjects are physical persons that can be found in the ANPO dataset, they could also be legal persons, which information can be found in another table, SOGG. We will thus divide our study in physical and legal holders, in order to facilitate the analysis. The only part that differs between the two cases is the first, where we associate to the utility holder its *IDR_SOG*. We will describe briefly the processes for the associations in both cases.

3.1 Physical persons

In order to associate the ACSOR identifiers of the subjects to the utility holders that are physical persons we need the Population Registry, and in particular the whole ANPO table. In fact, if we were to use the list of residents identified in Section 2, we would lose too much information. Indeed, a person could be the utility holder for an object even if he is not living in it. As a matter of fact, we have found that one of the utility owner died in 2013. This is an extreme case but it shows us that the holder could be any subject and not only the residents of Padova in the year 2021. However, what is certain is that if a subject is the proprietor of the bill, the object will be in a legal relation to it. Thus, after finding the *IDR_SOG* we will have a way to find the right object.
The key on which we performed the join of the two datasets, the whole ANPO table and the water bills, is the Tax ID Code, that is unique of each subject. After the joining, of the 95000 bills that have the value of the variable *tipo_sogg* set to *0* (physical persons) we associated 88000 ACSOR

identifiers. For the remaining rows, we tried to see if, by joining the water bills table with the ANPOS one, we could have more correspondences. We did not expect great results from this join, performed again on the Tax ID Code, because in theory the subject present in the historical Population Registry should be already in the ANPO table. As a matter of fact, we found only 8 new associations.
However, we still had around 7000 rows without the *IDR_SOG*. Since joining on the Tax ID Code was not enough, both because not all the subject in the Population Registry datasets have the value of *COD_FIS* not null and because there could be errors that do not make them comparable, we decided to go in a different direction and, after some data transformation, compare the name and surname of the subjects. In this way we could not ensure that the person associated was the right one (indeed for a name and surname we can have more than one person that has the same) but it is unlikely that two people that have the same appellative also are in relation with an object that is in the same address. This is not impossible, but it would be very rare. Thus when we will check if the object related to the subject is the one for which the bill is being paid, we will discard the information about the wrong person.
After all these processes, we have only 4000 bills for which we could not find an *IDR_OGG*. One of the reasons for these missing information is that the holder has never been registered in the Population Registry of the Municipality of Padova.

3.2 Legal persons

For the legal persons, that are identified by the variable *tipo_sogg* being set to *1*, we have a simpler process. In these rows, the variable *cf_tit_utenza* is not the Tax ID Code but rather the VAT number of the subject and the table in which we can find this information in the ACSOR database is the SOGG one. The number of bills of which the owner is a legal person is around 16000. After the join of the SOGG table with the water bills dataset, we find the *IDR_SOG* for 9000 of them. However, as opposed to the case of the physical persons, here the association is not unique. Indeed, since the legal persons are usually big companies or associations like a church, to the same VAT number can be linked many ACSOR identifiers. This will make the following analysis computationally more challenging, but the principles that are behind it are still correct.

After finding the *IDR_SOG* for the utility holders, the next step has been finding the correct object to which the bill refers to. As already mentioned in the case of the association between subjects and objects in Section 2, we need the table RUP as a link between the *IDR_SOG* and the *IDR_OGG*.
The usual challenge of identifying the correct object among all the ones that are related to a subject has now presented. For a more detailed explanation of the processes of comparison of the addresses the reader can refer to Section 1. We have created one new column *FLG_COR* in which we state the correctness of the association. If a bill has at least one row in which the *FLG_COR* is true, we eliminated all the other record that were referring to the same bill but are not a correct match.
Nonetheless, in this case we have a further information, that is that the water bill refers to one single precise object. There are many cases in which a bill is associated to more than one row with *FLG_COR* as true and so, in order to deal with this problem, we decided to remove some records based on the following criteria: if a bill is linked to more than one correct object (let's say the objects 1, 2 and 3), but another bill is associated to only one of those objects (for instance the object 2), since we know for sure that the object of the latter bill is the correct one for it, we can remove the object 2 from the association with the former bill. Since this concept is challenging to be explained in written words, we present the following tables for a more immediate understanding.

cod_id	*IDR_SOG*	*indirizzo*	*IDR_OGG*	*FLG_COR*	
1237	129381	Via J. Facciolati 7	1385	True	
1237	129381	Via J. Facciolati 7	24130	True	×
1237	129381	Via J. Facciolati 7	1207	True	

2903	34890	Via J. Facciolati 7	24130	True	✓

From this example we can see that the fictional subject *129381* is associated to many different *IDR_OGG* that have a correct correspondence with the address for which the bill is being paid. Instead, the *IDR_SOG 34890* has only one object related to it, that is consequently the right one for the bill. If the object *24130* is thus the one to which the bill *2903* is referring to, then it could not be the object of the bill *1237*. We will eliminate therefore that record because we know it is not the correct one.
The total number of objects of our list for which we found at least one bill

about them is around 79000.
The final aim of this part is to create the labels, using the knowledge that comes from the water bills. In order to distinguish between used and unused properties, we first need to calculate the variable on which we will set the threshold. In our initial dataset we have the variable *consumo* where we have the amount of water consumption in squared meters and the variable *n_mesi_fatt* that is the number of months for which the bill is being paid for. If we considered only the consumption, for bills that refer to few months we would think that they are below the minimum threshold. Instead, to have a more accurate understanding, we should look at the ratio between the consumption and the number of months.
As previously mentioned, the basic amount of water for each person in a day is between 50 and 100 litres. We will base our analysis on the lower end of this interval. If we considerate 50 litres of water for one person every day, it amounts to 18250 litres in a year, that is 18,25 m^3 per year and around 1,5 m^3 in a month. Thus we created a new column called *usage* where we set as *unused* all the objects that have the previously calculated ratio smaller than 1,5 and as *used* the ones with a higher ratio.
However, for the same object we could have more than one bill, and not necessarily all of them have the same *usage* status. Indeed, if a bill that refers to an object is paid for one month, when the consumption of water is low, and another bill refers to 11 months with a high *consumo* variable, the object is actually used. Thus if an object has at least one row in which the *usage* variable is set to *used*, we will consider it as so even if we had other rows with the status *unused*, since at least in a month of 2021 the property has been used.

4 Final dataset

We now have our list of objects with the knowledge about how many people live inside each object and their characteristics and the labels of *used* or *unused* in the rows for which we have found a correspondent bill. We are missing, though, some information about where the objects are spatially allocated. Indeed we cannot use the spatial identifiers because they do not have an order (e.g. an object with *sezione 25* is not necessarily near an object with *sezione 26*). Using some properties of ArcGIS, we where able to find the neighborhood for each object. The new column is called *cod_quart* and takes values from 1 to 6 to indicate the six different neighborhoods. We will see in next chapter if the information about where the objects are will

be a significant variable for our study.
The final list of objects, however, is still to be made. As seen in Section 1 of this chapter, there still are some *IDR_OGG* that have more than one row. In order to create the final dataset on which to perform the classification, we want each object to appear only once in the table.
In the IDCT table, one *IDR_OGG* can be associated to more than one set of spatial identifiers. Indeed, one of the reasons why we have the non-uniqueness of the objects is that in some cases many rows of CARTO, even if they refer to different spatial identifiers, have been mapped to the same *IDR_OGG*, especially when the only spatial identifier that is different is the *subalterno*. Thus, we decided to keep only one row for those objects that have many records that are equal in all the variables except for the spatial identifiers. For the same reasons, if we have that one *IDR_OGG* has many rows that have all the same informations (but the spatial ones) but there is only one row that has the label, we kept only the latter, since it is the record that provides us with the most knowledge.
After these operations we still have some *IDR_OGG* duplicated rows that, beside the spatial identifiers, also differ in other variables. For these cases we decided to keep, given the same quantity of information, the ones with the highest surface.
At the end, we find that the total amount of objects that were in Padova in 2021 amounts to 141138. Not all the rows in this dataset are labelled; indeed we have found a water bill only for about 79000 of the objects. The goal of the next chapter is thus to create a model that can accurately predict the labels that are already present and then classify the unlabelled objects. From now on, when we will refer to a dataset, it will be this final dataset. We will see that not all the columns will be necessary for the analysis, however they will be needed in a second moment to identify which are the objects that have been labelled as unused.

5 The classification

In this chapter we will present the model that we chose in order to classify the unlabelled objects. First we will start with the data exploration and feature selection, then we will focus on training different models and evaluate them based on some specific metrics and, after selecting the best model, we will predict the classes for the unlabelled dataset.
The dataset presents various columns, not all of which are important for the analysis. There are indeed some nominal variables that can be used in a second moment to identify the object. We will focus instead on the numerical and categorical columns.
In the first two sections we will consider both labelled and unlabelled data, since in order to have a consistent prediction we need all data to be transformed equally. Instead for choosing the best model we will only use the classified rows.

1 Data preprocessing and EDA

Data preprocessing is a critical step in data analysis since it impacts the quality and effectiveness of the models built on the data. We have already performed part of the process with the data cleaning implemented while creating this dataset. However, we still have to make sure of the non-presence of missing data, the conformity of values in each column and we have to deal with normalizing the raw data.
First, we have to treat the missing data. For the categorical variables we decided, given the number of None values, to create a new category called *Unknown*. Instead, for the numerical columns we opted for substituting the

missing values with the mean of the column.
Before proceeding with the normalization of our numerical data we studied the outliers and the distribution of the variables because, if we had not, the standardization would have been distorted. In the following boxplots we can see that we have a discrete amount of outliers. If we were to eliminate all outliers, however, we would incur in the loss of valuable information. Indeed, removing outliers without a careful analysis could lead to an oversimplified model that does not take into consideration exceptional events or unique patterns, leading to a conclusion that does not capture the complexity of the real-world phenomenon that is being studied. However, as we can see from Figure 1, there are a few cases in which the outliers are definitely different from the other values, thus we have decided to remove a few of the cases, that result in the deletion of 40 rows, that is a very low quantity compared to the total number of rows.
For instance, in the case of the *residenti_totali* column, we can clearly see that there is an outlier that is completely detached from the other values. It is possible in the real world to have a property in which there are more than 70 residents (e.g. a retirement home, given that also the *over 60* variable is very high), however for our study this data is completely out of range, thus we decided to remove the row with this datum.

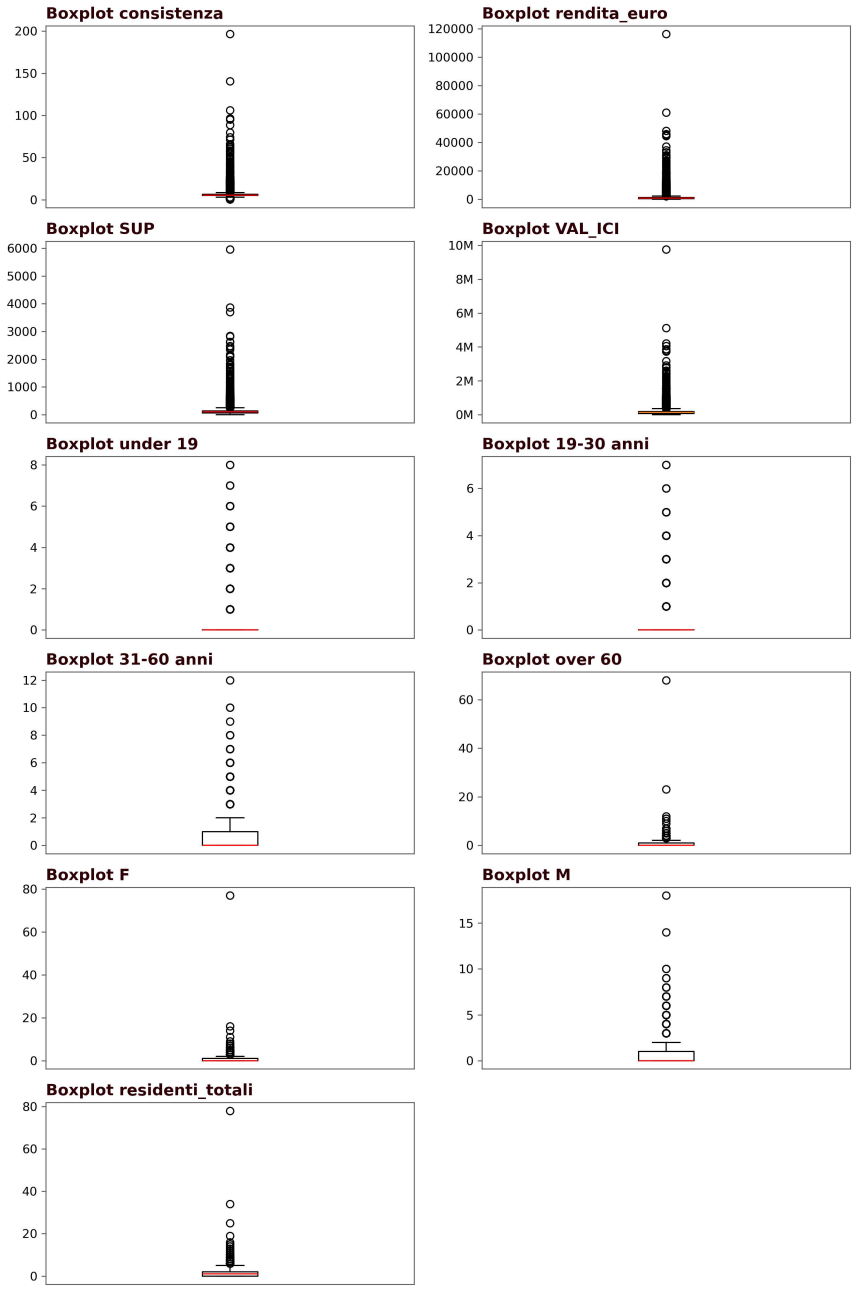


Figure 1: Study of the outliers

After the removal, we proceeded with the analysis of the distribution of the variables. This step is significant because it provides essential insights into the structure and characteristics of the data, allowing us to have a deeper understanding of the behaviour of our variables. It helps to identify patterns such as skewness or multimodality, enabling us to detect the presence of anomalies. Indeed, many algorithms require normally distributed data for optimal performance, thus understanding how our data deviate from normality helps us decide which transformation to implement.
We start by analyzing the numerical variables, that are also the ones on which we will perform the normalization. In Figure 2 we have plotted the histograms of the distributions. We can see that all the variables are highly right skewed. We notice, however, a difference between the first four features and the others: the ranges of values of the former is definitely higher than the ones of the latter. In both cases, furthermore, we have values that are 0, but in the variables related to the residents these cases are many more. In a certain sense, we could think about these features as categorical, however this way of thinking would be erroneous since the number of subjects that live inside an object cannot be thought as a category.
The method to normalize a skewed distribution is to make a logarithmic transformation of the data. Indeed, after performing this kind of alteration to the first four variables, the resulting distribution is undoubtedly more normal. On the contrary, we do not have much improvement for the features regarding the residents, thus we will leave them without the transformation.
Regarding the scaling, since the first four variables are now normally distributed, we used for them the python library *StandardScaler* that uses the formula

$$z = \frac{(x - \mu)}{\sigma}$$

where μ is the mean and σ is the standard deviation of the feature. This scaler assumes that the data are approximately normally distributed, thus it is the best fit for these four features.
For the remaining variables, instead, we cannot use the assumption of normality, therefore we have used the scaler *RobustScaler* that uses statistics that are less sensitive to outliers. The formula is the following:

$$z = \frac{x - med}{IQR}$$

where med is the median while IQR is the interquartile range ($75th$ percentile - $25th$ percentile).

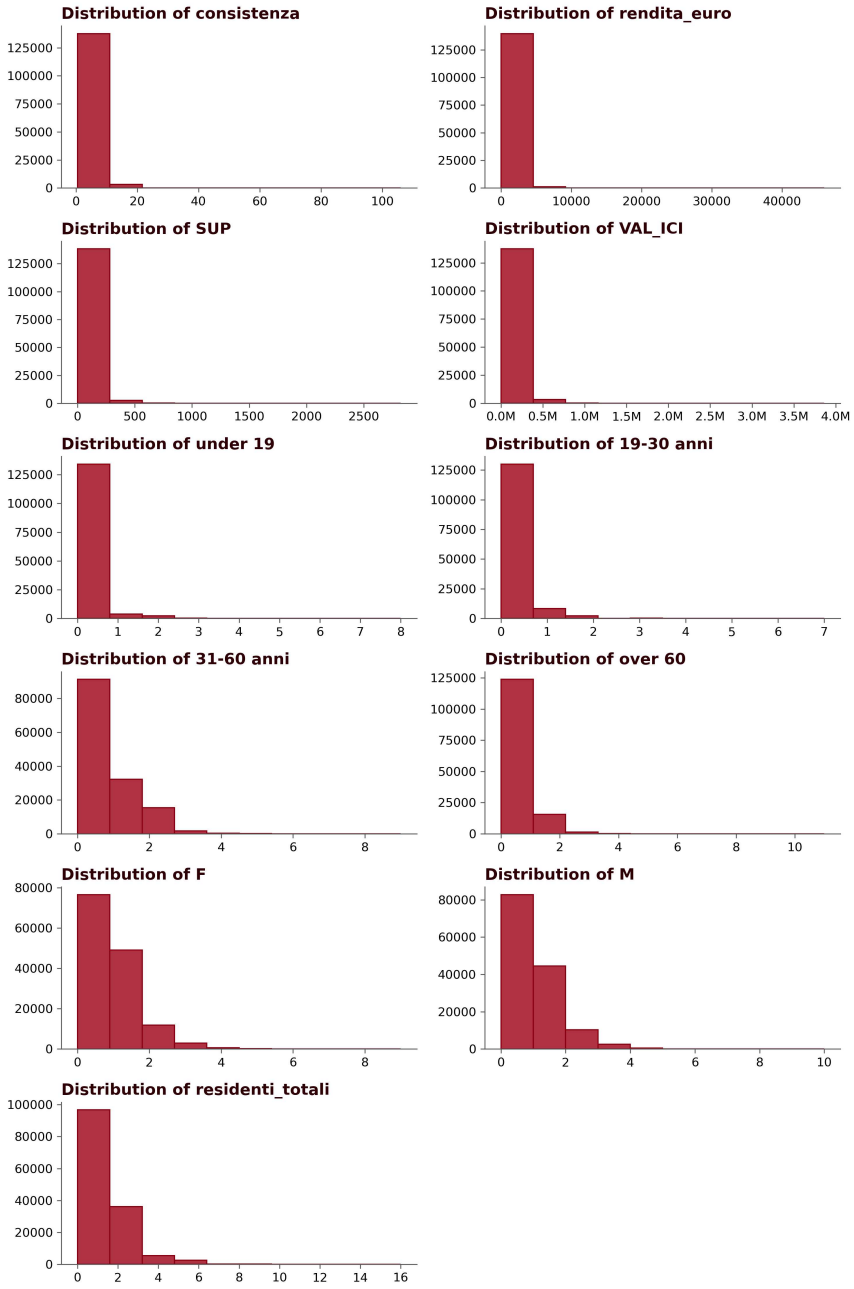
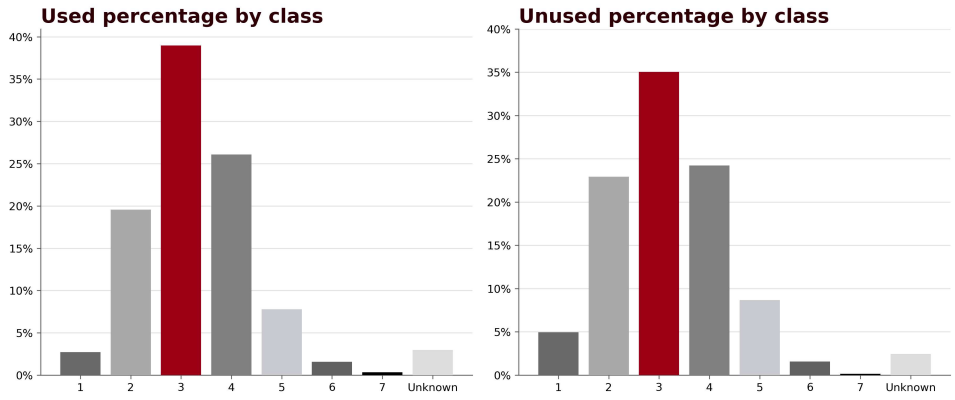


Figure 2: Distribution of the numerical variables

The next step is to handle the four categorical variables, that are the following: *categoria*, *classe*, *zona* and *cod_quart*.
Since the number of categories for each feature is at most 11, we decided to use the technique of one-hot encoding, using the pandas function *get_dummies*. With this function each variable is converted in as many binary variables as there are different values. This process is commonly used to convert categorical data into a form that can be provided to machine learning algorithms.
We now concentrate our Exploratory Data Analysis, that we will perform only on the labelled rows, on the categorical variables. We plotted the percentage of used (and then unused) objects for each categorical feature, in order to grasp which are the categories that have the highest frequencies. In Figure 3 we display one of these plots, that shows the distribution of the objects in each class. The cadastral class is an indicator of the productivity of a property, that is its ability to generate income. The class 1 indicates the lowest cadastral income, and the number of the class increases as the profitability and value of the property does.
We highlighted the class that has the majority of used (Figure 3a) and unused (Figure 3b) objects.



(a) (b)

Figure 3: Distribution of objects by class

As we can see from Figure 3, the highest number of used objects (and also of unused ones) is part of *class 3*.
We also report here in Figure 4 an interesting insight regarding how many unused objects there are in each neighborhood compared to the number of total objects in that neighborhood.

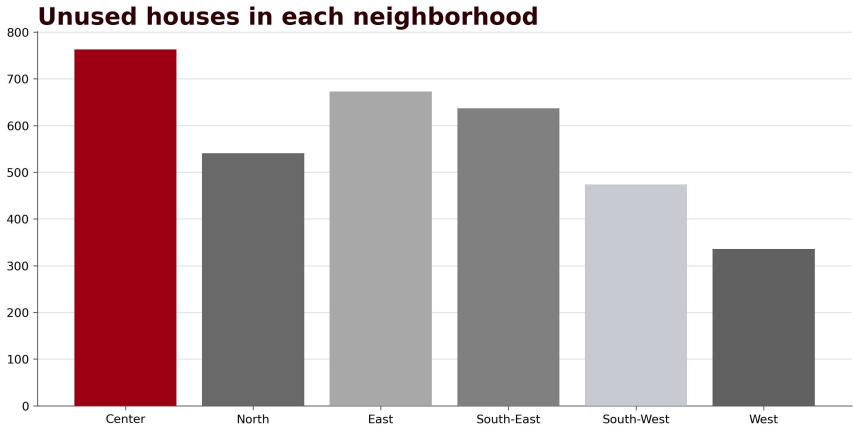


Figure 4: Number of unused houses per neighborhood

What we can understand from this plot is that the center is the neighborhood with the most unused houses that are already labelled. We will see if this behaviour is kept when all the objects will be classified.

2 Feature selection

Feature selection is a fundamental process that significantly improves model performance and interpretability. By selecting the most relevant features, we can reduce the dimensionality of the data, which helps to minimize the risk of overfitting, that is when a model performs well on the training set but poorly on unseen data. This process also improves computational efficiency by reducing the complexity of the model, making it faster and less cost requiring to train and use. Furthermore, feature selection eliminates irrelevant or redundant features that might introduce noise, leading to more accurate and reliable predictions.
The first step for a relevant feature selection is to study the correlation between variables. In fact, if two features are correlated, we can predict one from the other. Therefore, if two variables are correlated, the model only needs one, as the second one does not give additional information. For the numerical variables, we plotted the correlation matrix [7], that we report in Figure 5. Each cell in the matrix represents the correlation between two variables, measured using Pearson correlation, which is the most common method for assessing linear relationships. The Pearson correlation coefficient ranges from -1 to 1, where 1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship and 0 indicates no linear

relationship.

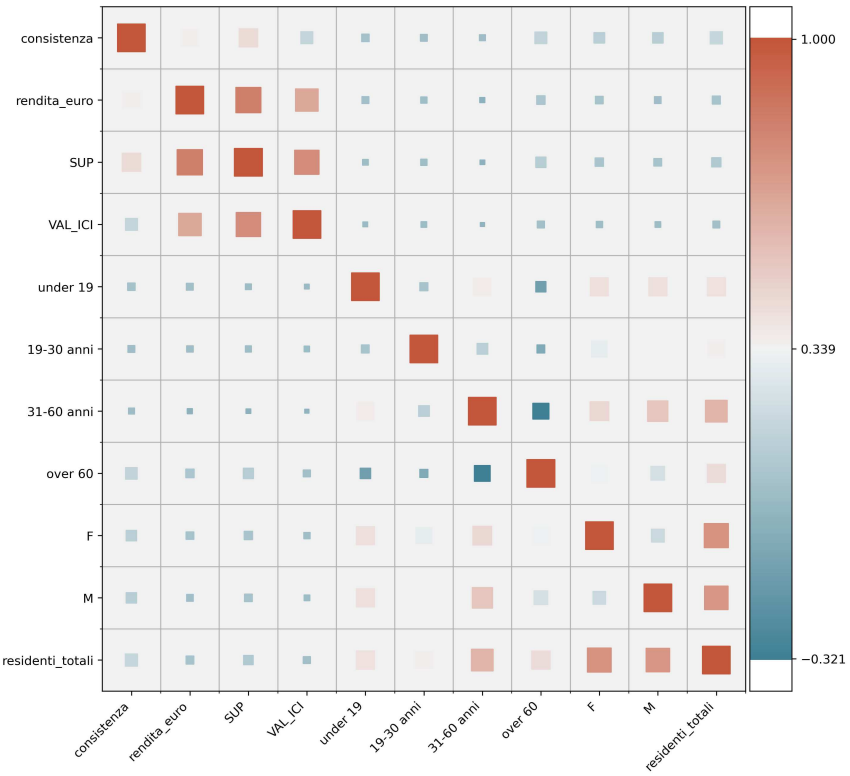


Figure 5: Correlation matrix

From the correlation matrix we can see that the numerical variables *rendita_euro*, *SUP* and *VAL_ICI* have a high linear relationship between each other, and we also have a noticeable positive linear relationship between the features *residenti_totali*, *F* and *M*. In the upcoming feature selection, we will need to consider these results and discard some of the correlated variables. With regard to the categorical features, the measure of association used is the Cramér's V statistic, which is derived from the chi-square statistic and ranges from 0 to 1, where 0 indicates no association between the variables and 1 indicates a perfect relationship. Unlike simple chi-square tests, which only tell whether there is a significant association, Cramér's V quantifies the strength of that relationship, making it easier to compare the associations between different pairs of categorical variables. From this statistic, no particular high association is being noticed.

We now look for multicollinearity, using the Variance Inflation Factor (VIF). Multicollinearity occurs when two or more independent variables are highly correlated, that means that they exhibit a strong linear relationship. This situation creates a problem because, when multicollinearity is present, it becomes difficult to determine the individual effect of each predictor variable on the dependent one, since their contributions are interlinked. This can lead to misinterpretation of the model's results.
The Variance Inflation Factor is a metric that is specifically used to detect this problem: it identifies the presence and intensity of multicollinearity among variables in a regression model. It measures how much the variance of an estimated regression coefficient is increased due to multicollinearity. VIF for a given variable X_i is calculated using the following formula:

$$VIF(X_i) = \frac{1}{1 - R_i^2}$$

where R_i^2 represents the proportion of the variance in X_i that can be explained by the other predictors. If R_i^2 is close to 1, this means that X_i is highly predictable from the other variables, indicating high multicollinearity. With the increase of R_i^2 towards 1, the denominator in the VIF formula approaches 0, making VIF more and more large. In practice, a VIF value of 1 indicates no correlation, while a VIF above 5 or 10 is considered to indicate a severe multicollinearity. Thus, one by one, we removed the features with a high VIF (we chose to eliminate those above 5) and in doing so we went from 40 variables to 27.
The number of features resulted is still high, thus we decided to calculate feature importance. This approach refers to all the different techniques that assign a score to input features based on how well they help predict a target variable.
We have used a Random Forest Classifier in order to calculate the scores for feature importance. This option is been guided by the definition of the Random Forest. Indeed, it is an ensemble of Decision Trees, each trained on a different random subset of the data, and they are hierarchical models that make splitting decisions based on the features. The choice of where to split is determined by how well a feature can separate the data into classes. Thus in its own definition, the Random Forest calculates a score for the importance of the features. In python, we can find the features importance score using the attribute *feature_importances_* of the RF Classifier. The calculation of the score is based on the Gini importance (also known as mean decrease in impurity), which measures the average reduction in Gini impurity. The

Gini impurity of a dataset is a number between 0-0.5, which indicates how often a randomly chosen element from the set would be misclassified if it was randomly labelled according to the labels distribution in the dataset. As a result of this technique, only 6 out of the 27 features are considered important, as we can detect from Figure 6.

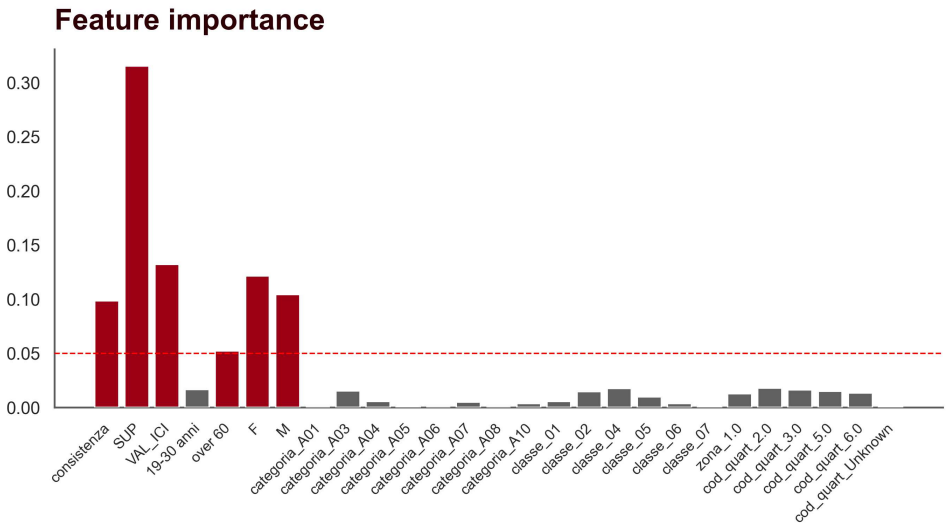


Figure 6: Feature importance scores

We can see that the feature that influences the most the splitting of the Decision Trees is the surface. As reported in Figure 6, the variables on which we will base our models are thus the following:

1. *SUP*
2. *VAL_ICI*
3. *F*
4. *M*
5. *consistenza*
6. *over 60*

We can notice that in these selected features we do not have both variables of the pairs of features that had a correlation higher than 0.8.

Something worth of notice is that we tried training our models with a larger subset of variables, but the results did not improve. Thus we decided to keep only those found with the feature importance technique.

3 Handling imbalanced classes

An important characteristic of our dataset is that the classes are heavily imbalanced, that means that one class significantly outbalances the other, as we can grasp from Figure 7:

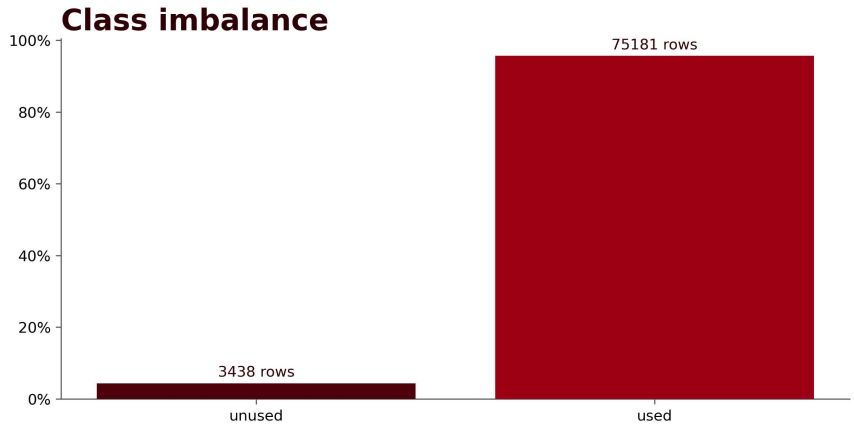


Figure 7: Imbalance of the classes

Indeed, we have that approximately only 4% of the labelled objects are unused.
Class imbalance is a noteworthy challenge: it can lead to biased models that perform well on the majority class but poorly on the minority class, since the algorithm does not have enough data to learn the patterns present in the minority class.
One of the most used techniques to handle class imbalance is resampling, both in the forms of oversampling or undersampling. The former involves increasing the number of instances in the minority class by repeating the samples, while the latter entails the removal of a certain number of samples from the majority class. Even though these two strategies accomplish to balance the classes, somehow they risk introducing new issues. With undersampling we could incur in the model missing out on some important patterns that were in the removed samples. Instead with oversampling,

given the repetition of the samples, the training could be slowed down and there could also be an overfitting in the model [8].
We thus chose to not perform one of these techniques, but to use class weights. We will give different weights to both the majority and minority classes: the goal is to penalize the misclassification made on the minority class by giving it a higher class weight, and at the same time decreasing the weight for the minority class, ensuring that the model pays more attention to the underrepresented group. In this way the model improves its ability to generalize and accurately predict outcomes across all classes [9].
Most of the sklearn classifier modelling libraries have a parameter in which we can state the classes weights. If there is no specification, we have no particular weights associated to the classes; we can instead use the *balanced* option that automatically regulates the weights or we can also manually set a pair of weights.
Another challenge related to the imbalance of classes is splitting the data into a train and test set. Indeed, if we randomly divide our labelled data into train and test without any consideration, since we have many more samples of the *used* class, we could face the situation in which we have none of the *unused* examples in the training set, making the learning of the model completely biased. In order to avoid this, we set the *stratify* parameter of the *train_test_split* function of sklearn to take into account the labels. This parameter ensures that the split respects the distribution of the target labels. This guarantees that the training and test sets are stratified, meaning they will have the same proportion of classes as the original dataset.

4 Model training and evaluation

Before describing the different models that we tried and selecting the best one, we focus on the metrics on which we evaluated them. In our case of heavily imbalanced classes, where *1* is associated to *used* and *0* to *unused*, we cannot base our evaluation on accuracy: indeed, if we were to classify all objects as *used*, we would still have high accuracy since most of the instances belong to that class, however the result would be useless for our study. Instead, we want to minimize the *type I error*, i.e. the False Positives: this is because if we classify an object as unused and in truth it is used (False Negatives), after a physical inspection of the property we could asses that it is indeed used. On the contrary, if we perform a misclassification by stating that an unused object is actually used, we would never perform a check on that house and thus we would never know that it is not used.

In order to minimize the false positives we should maximize the specificity that has the following formula:

$$specificity = \frac{TN}{N}$$

where TN are the True Negatives while N are the actual Negatives. However, having too many objects to control would be pointless; we still need to have a high accuracy: in fact, if we were to set all labels to *unused* we would have a high specificity but a low accuracy. The formula for accuracy, that detects how accurate and therefore how many errors the model performs, is the following:

$$accuracy = \frac{TP + TN}{P + N}$$

where TP are the True Positives, P are the actual Positives and the other two addends are the same as for specificity.
Furthermore, we want to not being forced to control the truth of too many objects, thus we want also to minimize the False Negatives, using the sensitivity score, also known as recall, that is specified by the following formula:

$$recall = \frac{TP}{P}$$

Thus, we look for a model that balances specificity, accuracy and recall. In order to evaluate the models on this metric, we will search for the one that maximize the following:

$$acc\_spec\_rec\_balance = \frac{1}{3}accuracy + \frac{1}{3}specificity + \frac{1}{3}recall$$

We will still look at how the model performs with regard to the other metrics, but they will have a smaller influence on our selection of the best one. In particular we will pay attention to the F1 score, that is a weighted average of precision and recall. The standard F1 score is defined to address the positive class; nonetheless, our study actually focuses on well predicting the negative class, thus we will use the formula:

$$F1 = \frac{2TN}{2TN + FN + FP}$$

Since our analysis is focused on a binary classification, we have tried some models that achieve this goal. In particular, our study revolves around the Logistic Regression, the Random Forest Classifier, the K-Nearest Neighbour Classifier and the Stochastic Gradient Descent Classifier. Each of these models has its own strength and weaknesses related to our objective. Our aim is thus to select which model performs the best in our case, based on the chosen metrics.
As predicted in Section 3, if we do not take into consideration some weights for the classes, the models perform poorly. In order to have a concrete evidence, we trained a Logistic Regression on the training data without setting any weight and, as expected, the model classified all the test samples as *used*, returning high accuracy and precision scores but a specificity score of 0.
For each model, except for the K-NN, we trained both the automatic *balanced* option for the weights, that uses the y values to automatically adjust weights inversely proportional to class frequencies in the input data [10], and a model with the best weights, found through a grid search, that maximizes the metrics.
Grid search is a technique used to systematically search through a predefined set of hyperparameters to find the optimal combination for a model. Instead of randomly choosing values or tuning them manually, the grid search systematically explores a predefined set of values, evaluating the model's performance for each combination. The process involves training the model multiple times, each time with a different set of weights, and measuring its performance using the scoring metric that we previously decided. The combination of weights that has the best performance on the training set is then chosen as the best one for the model.
For the evaluations of our models we fix a random state, therefore the confusion matrices reported below reflect this specific state. However, even without fixing the random state, the results would be similar.

4.1 Logistic Regression

Differently from Linear Regression, which predicts continuous outcomes, Logistic Regression estimates the probability that a given input belongs to a particular class. It does this by applying a logistic function (also known as the sigmoid function) to the linear combination of the input variables, producing an output between 0 and 1, which can be interpreted as the probability of the positive class. The logistic function is defined as follows:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)}}$$

where $P(y = 1|X)$ is the probability that the output y is 1 given the input features $X = [X_1, X_2, ..., X_n]$, β_0 is the intercept term and β_1, β_2, ..., β_n are the coefficients of the input features.
Logistic Regression is particularly powerful because it can handle both continuous and categorical variables and provides interpretable coefficients that indicate the strength of the relationship between predictors and the outcome. Its simplicity and interpretability make it a frequently used model for solving binary classification problems.
As previously mentioned, we have trained the Logistic Regression with different pairs of class weights. In figure 8 we report the grid search for the best weights for the Logistic Regression with the scope of maximizing the *acc_spec_rec_balance* score.

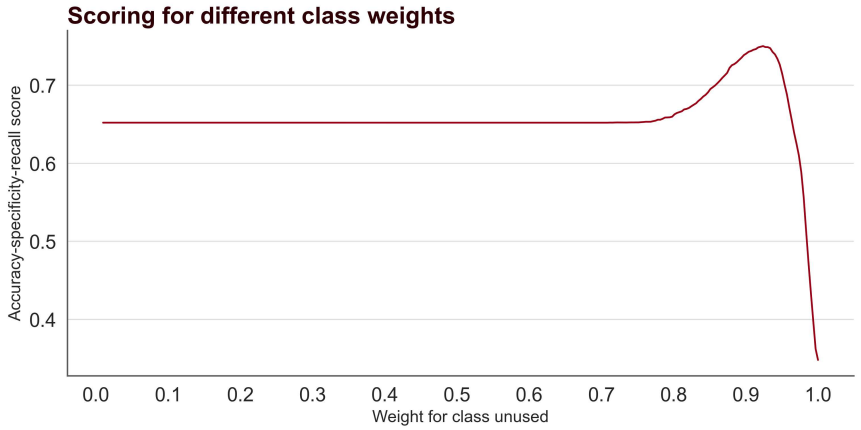


Figure 8: Grid search Logistic Regression

From this plot we can see that as we increase the weight for the *unused* class, also the balanced score between accuracy, specificity and recall increases, until it reaches a maximum point and then it starts decreasing. That maximum is the weight for the negative class that will give us the best performance for the Logistic Regression on the aforementioned metric.
We also report, in Figure 9, the three different confusion matrices that describe the predicted values for the test set. The matrix 9a refers to the Logistic Regression trained without weights, Figure 9b is the confusion ma-

trix for the model trained with the *balanced* option while 9c refers to the Logistic Regression with the best weights found through the grid search.

(a) (b) (c)

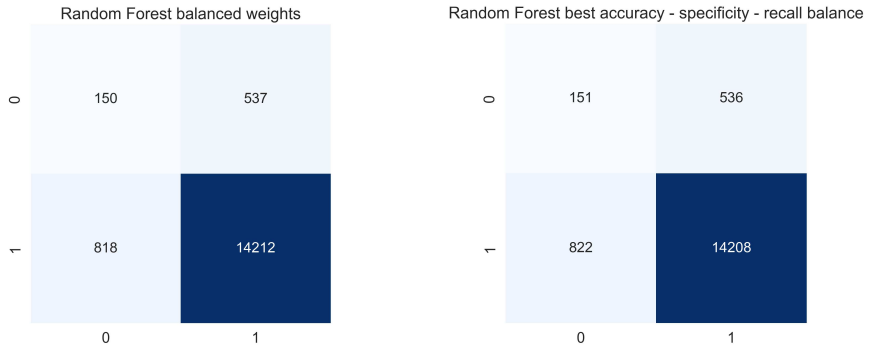
Figure 9: Confusion matrices Logistic Regression

The best result for the Logistic Regression regarding the metric is indeed the model with the weights found through the grid search. As we will see also in Figure 14, the score for the balanced model is around 0.697 while for the model with the best weights it is 0.762. Furthermore, we can see that the difference between the two models stands in the number of errors, both of first and second type. Indeed the balanced model is much less accurate than the other one.

4.2 Random Forest Classifier

The Random Forest classifier is an ensemble learning method that is very effective for binary classification problems. It creates many decision trees during training, where each tree is built on a random subset of the data and a random subset of features. The final prediction for a new data point is made by joining the predictions of all the individual trees, typically using a majority voting system. This approach is used in order to decrease the risk of overfitting that can occur with individual decision trees, leading to a more robust model. Random Forest is particularly powerful because it can handle large datasets with high-dimensional feature spaces and can manage both continuous and categorical variables.
We have trained two Random Forest Classifiers, one with the balanced option and the other with the weights estimated through a grid search, with the aim of maximizing the usual *acc_spec_rec_balance* metric. We report in

Figure 10 the confusion matrices of the two models.



(a) (b)

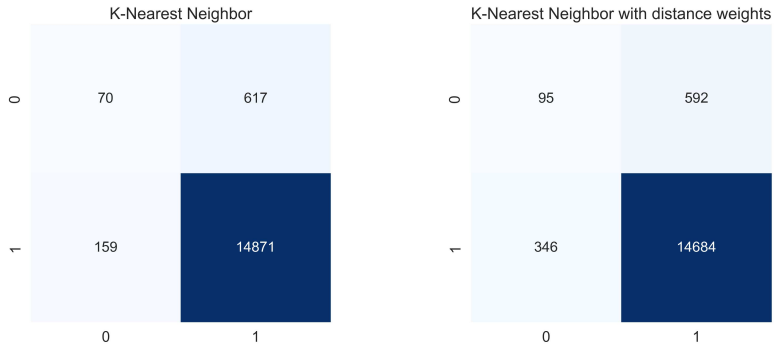
Figure 10: Confusion matrices Random Forest Classifier

Even thought the accuracy of both these models is high, they have a low specificity score. As a matter of fact, as we can observe from the confusion matrices, we have more False Positives than True Negatives, and this means that the *unused* class is not well predicted. For this reason, the Random Forest Classifier is not be the best model for our case.

4.3 K-Nearest Neighbor

K-Nearest Neighbors (KNN) is a simple but powerful algorithm used for binary classification that depends on the idea of similarity between data points. In KNN, when a new data point needs to be classified, the algorithm looks at the k closest data points and assigns the most common class among these neighbors to the new point. The choice of k is fundamental: a small hyperparameter could make the model sensitive to noise, and thus overfit, while a large k may lead to underfitting. Furthermore, it is advised to use an even number for k in order to avoid the situation in which we have the same amount of neighbors of different classes. KNN is non-parametric, and this means that it does not assume any underlying distribution of the data, thus it is flexible and applicable to various types of data. However, one disadvantage about KNN is that it can be computationally expensive for large datasets since it requires calculating the distance between the new data point and all other points in the dataset.
In our study, we trained both a simple KNN and a K-Nearest Neighbor Classifier with the parameter *weight* set to *distance*. This specification weights

points by the inverse of their distance; closer neighbors of a query point will have a greater influence than neighbors which are further away [10].
In Figure 11 we report the confusion matrices for both models.



(a) (b)

Figure 11: Confusion matrices KNN

Similarly to what happened for the Random Forest Classifiers, we have a high accuracy but a very low specificity. These models will therefore not be the best for our case, as we will see in Figure 14.

4.4 Stochastic Gradient Descent Classifier

The Stochastic Gradient Descent (SGD) Classifier is an efficient algorithm that is commonly used for binary classification problems, in particular when dealing with large datasets or high-dimensional data. The SGD Classifier is an implementation of a linear model, such as logistic regression or support vector machines (SVM), that employs the Stochastic Gradient Descent method for optimization. The difference from the traditional Gradient Descent, which computes the gradient of the loss function using the entire dataset, is that SGD, given a function $f(\theta)$ to minimize, updates the parameters θ iteratively by calculating the gradient based on a single training example or a small set at each step, according to the rule:

$$\theta_{t+1} = \theta_t - \eta \nabla f(\theta_t; x_i, y_i)$$

where θ_t represents the parameters at the t-th iteration, η is the learning rate and $\nabla f(\theta_t; x_i, y_i)$ is the gradient of the objective function computed

using a single training example (x_i, y_i).
For the classification, the objective function $f(\theta)$ is typically the loss function. This method significantly increases the learning process rate and reduces the use of memory. Indeed, since each update only requires a single example or a small batch of samples, this method is able to handle large datasets efficiently.
In our case, we trained two different SGD Classifier, one with the weight parameter set to *balanced* and the other with the class weights found through a grid search to maximize our goal metric. We have tried different types of *loss* parameter, that is the value that decides which loss function has to be used. The results that we provide in Figure 12 refer to a linear support vector machine with SGD method for optimization, which gave us the best results.

(a) (b)

Figure 12: Confusion matrices SGD Classifier

From these matrices we can notice that the SGD with balanced weights has a lower accuracy than the other model, while it has a higher specificity. However, as we will observe in Figure 14, the metric on which we base our study, the *acc_spec_rec_balance* score, is higher in the SGD with weights from the grid search than in the balanced model.
We are interested in studying more in details the last mentioned model. In particular, we want to investigate which variables have more importance in the classification made by the model thus, in order to do so, we look at the weights assigned to the features, reported in Figure 13. In order to correctly understand the results, we should pay attention to the absolute values of the coefficients. The higher is the magnitude, the more influential is the feature.

	Coefficients
SUP	-0.123222
VAL_ICI	0.141338
consistenza	0.045194
F	1.528236
M	1.607276
over 60	0.047780

Figure 13: Feature weights using best SGD Classifier model

We can notice that the model assigns a higher weight to the variables related to the sex of the residents, that therefore will be the most influential features. We will see in Section 5 why this result will majorly impact the prediction of the unlabelled rows.

5 Prediction on unlabelled data

The last step is to perform the classification on the unlabelled objects. We need to understand in which category the remaining rows that are unlabelled fall; however we have to consider that the results of the classification cannot be immediately confirmed nor refused, since there is no actual evidence of the state of usage of the object at the moment.
We thus will perform the classification using our best knowledge, that is using the model that performed better with respect to the goal metric among the ones that we have previously mentioned in Section 4.
In Figure 14 we report a model performance comparison with respect to the *acc_spec_rec_balance* score, that is our core metric, and also on the F1 score. The highlighted model is the one that has the highest score performance.

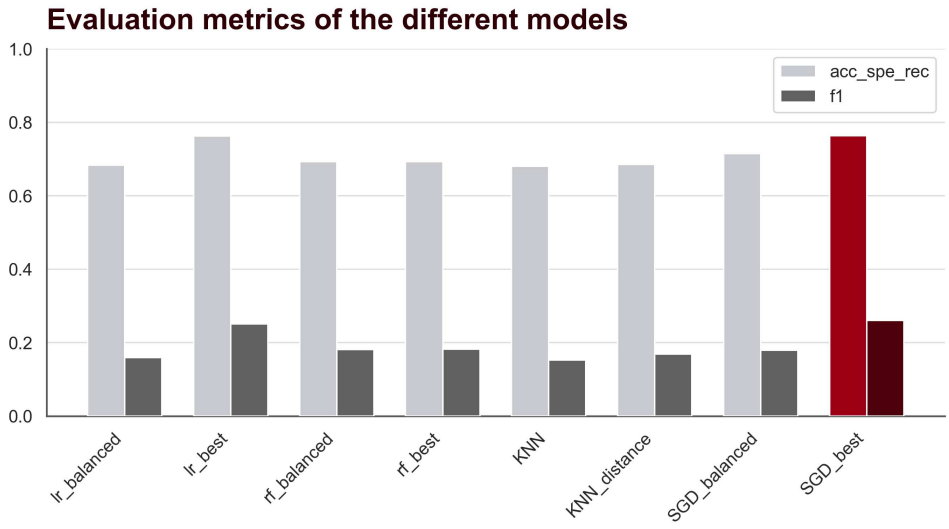


Figure 14: Models performance on different metrics

The *acc_spec_rec_balance* scores for all the models are similar, but there are two models that stand out, that are the Logistic Regression and the SGD Classifier, both with the class weights found through the grid search. These two models give very similar results, however we have chosen the latter one since it performs slightly better both on the metric score and on the F1 score.

The best model for our analysis is then the Stochastic Gradient Descent Classifier with the class weights found through the grid search. We have thus used this model to predict if the unlabelled objects are *used* or *unused*. The result that we obtained is peculiar: all the unlabelled objects that have at least one resident have been predicted as used, while the ones with the variable *residenti_totali* set to 0 have been assigned the *unused* label. This classification, however, is not completely surprising since, as seen in Figure 14, the features that are most influential for this model are the ones related to the sex of the residents, and therefore this model is highly based on the presence or absence of inhabitants.

In this way, we get a totality of around 48000 unused properties, that is slightly more than $\frac{1}{3}$ of the total amount of objects. Even if high, this result could be explained by the possibility that an object not associated with a water bill, and thus previously part of the unlabelled set, might not be connected to any water supply and be therefore likely unused.

An interesting aspect, useful for the Municipality of Padova, is to detect in

which part of the city the unused objects are situated. We have created, using the software ArcGIS, a map of the city of Padova divided in neighborhoods, that is coloured based on the density of the presence of not utilized objects. This map is reported in Figure 15.

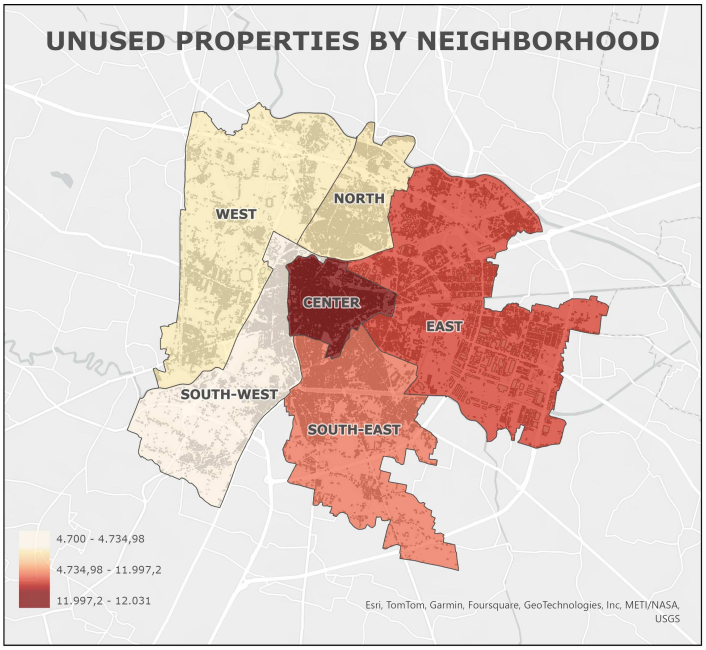


Figure 15: Density of unused properties by neighborhood

Comparing the results of Figure 15, where the totality of objects is represented, with those of Figure 4, where only the unused properties that were connected to the water are taken into account, we can see that, for the majority, the ranking of the neighborhoods that have the most unused houses is unchanged. The only difference is that, before inserting the previously unlabelled objects, the West neighborhood was the one with the least amount of unused properties while now that place is occupied by the South-West area.
We can notice that the Center zone has remained the one with the highest number of unutilized objects. One of the reasons of this condition could be associated with the higher prices of the houses that are situated in that area.

Another interesting insight is comparing the proportion of unused houses over all the properties in a neighborhood: indeed, it would be logical to think that an area with more houses has a greater probability of having more not utilized objects. Thus, the correct way of studying this phenomenon is looking at the percentage of unused properties. This situation is represented in Figure 16.

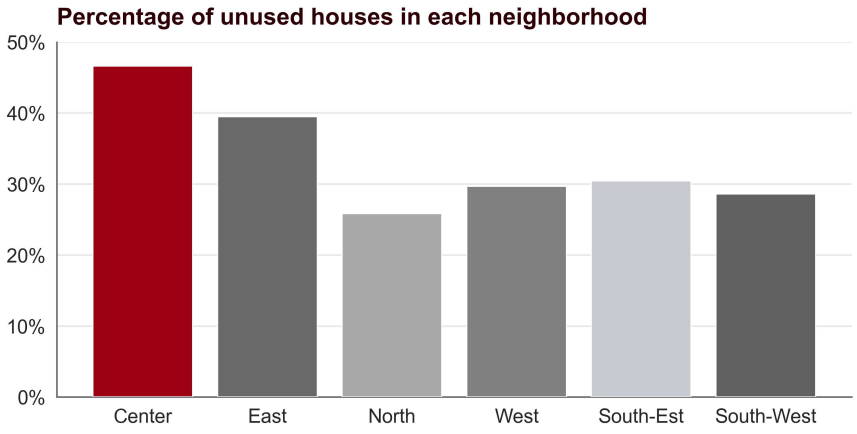


Figure 16: Percentage of unused objects in each neighborhood

We can see that the South-West neighborhood is not the last anymore. This means that the density of houses in that particular area is low, thus even a smaller amount of unused objects results in a higher percentage with respect to all the properties present.
In order to physically check if a property is really not utilized, each single unused object can be identified by the nominal variables that we have not used for the models but that are key in order to understand where the property is positioned. Indeed, using the spatial identifiers in union with the address allows us to precisely allocate the object.

6
Conclusions

As a result of this study, we have gathered some useful insights for the Municipality of Padova about the problem of unused properties in the city. First of all, the connections among the different tables will be functional for further studies, not only on the subject of this dissertation. Indeed, the different processes of data preparation have led to some cleaned and accurate tables that can be used as bases for other Municipality's projects.
Furthermore, with regard to the issue subject of this study, we have found that, of the 141000 properties that were present in the city of Padova in 2021, about 48000 have been labelled as *unused*. Even though we are certain of the state of usage of only around 3500 objects, for which we have found the associated water bill that states the consumption of water, we were able to classify the other properties using a machine learning model, in particular the Stochastic Gradient Descent Classifier with ad hoc class weights. This amount of unused objects, although high, can be explained by the fact that, if we do not have a water bill associated to a property, it is likely that the object is not connected to water, thus it is very improbable that it is being used.
The model employed, for the classification of the object, relies heavily on the presence or absence of residents, that we have found by connecting the Population Registry with the Land Registry. A suggestion for future researches and to refine the result discovered could be to use other sources of information that can enhance the knowledge of the problem, as for example the data about the Waste Tax, that was not available at the time of this study. More sources of information could give the model other important features to learn from, arriving to a more accurate result.

Furthermore, more source of knowledge could bring a deeper understanding on why some properties are classified as *unused*. For instance an useful information could be knowing if the unused objects are actually in the process of being renovated. Indeed, a property that is undergoing a renovation could be only momentarily unutilized, however its classification as an unused object could negatively influence our result.
Another important insight that we have found in our study is that most of the unused properties are allocated in the Center area; this result, however, does not come as a surprise knowing that typically the houses in the Center zone of a city are more expensive than those in the peripheral areas.
In recent years the Municipality of Padova has already developed some policies in order to address the problem of unused houses, as for instance giving some tax advantages to the owners that rent their properties to students. However, with the results discovered from this study, some more targeted measures can be taken. As a matter of fact, the unused properties can be spatially allocated and also associated to their owner, thus the Municipality could theoretically even contact each landlord directly.
This research was intended to be a starting point for even more detailed studies and evidence of how public bodies can also use data analysis for problems that concern citizens and municipalities.

Appendix

In this appendix we report the description of the important variables of the datasets that we presented in Chapter 3 and that we used throughout our study.

1 IDCT

- *COD_ENT* : the code of the municipality; for Padova it is G224.
- ***IDR_OGG*** : the unique identifier of an object in the ACSOR environment.
- *STA* : the sequential number that identifies the different stages of the object.
- ***SEZ*** : the original name is *Sezione catastale*. It is the first of the four spatial identifiers of an object.
- ***FOG*** : the original name is *Foglio catastale*. It is the second of the four spatial identifiers of an object.
- ***NUM*** : the original name is *Mappale catastale*. It is the third of the four spatial identifiers of an object.
- ***SUB*** : the original name is *Subalterno catastale*. It is the last of the four spatial identifiers of an object.

2 UIU

- *COD_ENT* : the code of the municipality; for Padova it is G224.
- ***IDR_OGG*** : the unique identifier of an object in the ACSOR environment.
- *STA* : the sequential number that identifies the different stages of the object.
- ***CAT*** : the cadastral category, that identifies the type of the object.

- *CLS* : the cadastral class; it is a criteria to distinguish objects belonging to a certain cadastral category in relation to the finishing, the position, etc.
- *CNS* : the cadastral extension. It is a measure of the size of the object specifically of the land registry.
- *SUP* : the surface area of the object in squared meters.
- *ZON* : the census area; the city is divided in census areas in which the environmental and socio-economical characteristics are similar.
- *IMP_REN* : the cadastral income, that is the value, for tax purposes, of real estate properties on the basis of the provisional rate.

3 INDO

- *COD_ENT* : the code of the municipality; for Padova it is G224.
- ***IDR_OGG*** : the unique identifier of an object in the ACSOR environment.
- *PRG* : the progressive number that characterizes each row related to the same object.
- *TIP_IND* : a binary variable that can be either P, if the address is the principal one, or S, if the street name is secondary.
- ***DEN_VIA*** : the name of the street.
- ***NUM_CIV*** : the house number.
- *ESP_CIV* : if the house number has also a letter, the latter is reported, otherwise the field is void.
- *NUM_INT* : the number usually located on the door of flats in a flat complex.
- *PIANO* : the floor number.

4 MW_RICERCA_FUSIONI

- *COD_ENT* : the code of the municipality; for Padova it is G224.
- ***ORIG_IDR*** : the identifier of the object in the ACSOR environment that has been merged to the the DEST_IDR.
- ***DEST_IDR*** : the new identifier of the object in the ACSOR environment after the merge.
- *DAT_ELAB* : the date in which the merge procedure took place.

5 CARTO

- ***categoria*** : the cadastral category, that identifies the type of the object.
- *classe* : the cadastral class.
- *codice_comune* : the code of the municipality.
- *consistenza* : the cadastral extension.
- ***data_inizio*** : the start date of validity of the object.
- ***data_fine*** : the end date of validity of the object. If the object is still valid, it has been set to 99991231 or 00000000.
- ***indirizzo_catasto*** : the address written with the rules of the land registry.
- ***indirizzo_comune*** : the address written with the rules of the Municipality office.
- ***sezione*** : the first of the four spatial identifiers of an object.
- ***foglio*** : the second of the four spatial identifiers of an object.
- ***mappale*** : the third of the four spatial identifiers of an object.
- ***subalterno*** : the last of the four spatial identifiers of an object.
- *zona* : the census area; the city is divided in census areas in which the environmental and socio-economical characteristics are similar.

6 ANPO / ANPOS

- *COD_ENT* : the code of the Municipality; for Padova it is G224.
- ***IDR_SOG_STL*** : the unique identifier of a subject in the Population Registry environment.
- *DEN_SOG* : the surname and name of the subject.
- *DAT_NSC* : the date of birth of the subject.
- *DEN_CMN_NSC* : the name of the city of birthplace of the subject.
- *SGL_NAZ_NSC* : the initial of the country of birthplace of the subject.
- *SEX* : the sex of the subject.
- ***COD_FIS*** : the Tax ID Code, that is unique for each subject.
- *DAT_DEC* : the date of death of the subject. If the person is still living, the date is set to *00000000*.
- ***DEN_VIA*** : the home address of the subject.
- *NUM_CIV* : the house number related to the home address of the subject
- ***DAT_INI_RES_VIA*** : the date in which the subject has started living in the home address.
- ***DAT_FIN_RES_VIA*** : the date in which the subject has stopped living in the home adress. If the subject is still living there, it has been set to *99991231* or *00000000*.
- *DAT_VAR* : the date in which the row has been changed and a new record has been created. This means that in this date the row has passed from ANPO to ANPOS.

7 SOGG

- *COD_ENT* : the code of the Municipality; for Padova it is G224.
- ***IDR_SOG*** : the unique identifier of the subject in the ACSOR environment.

- *DEN_SOG* : the surname and name of the subject.
- *COD_FIS* : the Tax ID Code, that is unique for each subject.
- ***PAR_IVA*** : the VAT number, that is unique for each subject.

8 RESS

- *COD_ENTE* : the code of the Municipality; for Padova it is G224.
- ***IDR_SYS_STL*** : the code for the operational source. We will need only the rows in which this value is *ANAPOP*.
- ***IDR_SOG_STL*** : the unique identifier of the subject in the source described by the *IDR_SYS_STL* value.
- ***IDR_SOG*** : the unique identifier of the subject in the ACSOR environment.

9 RUP

- *COD_ENTE* : the code of the Municipality; for Padova it is G224.
- ***IDR_SYS_STL*** : the code for the operational source.
- *ID_REL* : the unique identifier of the relation.
- ***IDR_SOG*** : the unique identifier of the subject in the ACSOR environment.
- ***IDR_SOG_STL*** : the unique identifier of the subject in the source described by the *IDR_SYS_STL* value.
- ***IDR_OGG*** : the unique identifier of the object in the ACSOR environment.
- *TIP_REL* : the type of relation between subject and object (can take values *PP* for ownership or *UT* for usage)
- *DAT_INI* : the start date of validity of the relation.
- *DAT_FIN* : the end date of validity of the relation. If the relation is still effective, it has been set to 99991231 or 00000000.

10 Water bill

- *anno* : the year to which the bills are referred to.
- *cod_cat* : the code of the Municipality; for Padova it is G224.
- *cf_erogante* : the Tax ID Code of the providing company.
- ***cf_tit_utenza*** : the Tax ID Code of the subject that is the utility holder.
- ***tipo_sogg*** : a binary variable that identifies the type of the subject, where 0 stands for natural person while 1 is a subject different from a natural person (i.e. a legal person).
- ***dati_anag*** : the personal data of the subject, like the name, surname, date and city of birth (if they are of type 0) or the name of the society (for the type 1).
- *cod_id* : the identifier code of the bill
- *tipo_ut* : the classification of the utility. The categories are 1, that is domestic use with residency in the place for which the bill is being paid, 2 , that is domestic use with residency different from the place for which the bill is being paid, and 3, non domestic use.
- ***indirizzo*** : the name and number of the home address of the object of the bill.
- *segno* : the sign of the invoice; + is positive while - is negative.
- *fatturato* : the ammount of the invoice, in euros.
- ***consumo*** : the consumption of the water in cubic meters.
- ***n_mesi_fatt*** : the number of months for which the bill is being paid for.

References

[1] *Urban challenges, housing solutions*, Feantsa, 2022

[2] *Sono oltre 10 milioni le case inabitate in Italia*, Openpolis, 2023

[3] *https://municipia.eng.it*

[4] *The Italian Land Registration System*, Agenzia delle Entrate, 2022

[5] *Methodology and Evaluation of Population Registers and Similar Systems*, United Nations, 1969

[6] *The Human Right to Water and Sanitation*, World Health Organization

[7] *Better Heatmaps and Correlation Matrix Plots in Python*, Drazen Zaric, 2019

[8] *Handling Class Imbalance by Introducing Sample Weighting in the Loss Function*, Ishan Shrivastava, 2020

[9] *How to Improve Class Imbalance using Class Weights in Machine Learning?*, Kamaldeep Singh, 2023

[10] Scikit learn documentation

Acknowledgements

First and foremost, I want to thank the Municipality of Padova for giving me the opportunity to carry out such an important and interesting project. In particular I want to express my gratitude to Ing. Alberto Corò and Ing. Pietro Fontolan for guiding me in the complexities of the real world.
A large thank you goes also to my supervisor, Professor Alberto Roverato, for always giving me important suggestions to reflect on and advising me on new directions to take.
A big shoutout also goes to my family, by blood and otherwise, for consistently being by my side and, without fail, believing in me.
I would also like to thank my friends, who have accompanied me through another two years of ups and especially downs. In particular my love goes to Giacomo, Alice, Francesco, Diletta and Matteo. Thank you for always being by my side, I hope to have you for one hundred other years to come.
And finally to my fellow classmates, with whom I have studied so many hours but also laughed quite as many. I hope for each one of you to achieve all the goals that you have set for yourself.