INSTITUTO POLITÉCNICO
DE BRAGANÇA

Machine Learning for Data Science

Authors:
Joran Andringa

André Felipe Bianek da Silveira

Roberto Chávez Trujillo

Ayoze Jesús Guanche Núñez

Barbare Pipia

Priscila Reyes Torres

Contents

1 Problem Description 2

1.1 Dataset Variables . . . 2

1.2 Classification Objective . . . 2

1.3 Project Significance . . . 2

1.4 Dataset Characterization: Exploration and Initial Analysis . . . 3

2 Methodology 4

2.1 Exploratory Data Analysis (E.D.A) . . . 4

2.1.1 Graphs and Descriptive Statistics . . . 4

2.1.2 Other Descriptive Graphs . . . 5

2.1.3 Stats by Installation Zone . . . 7

2.2 Data Preprocessing . . . 8

2.2.1 Completing the Dataframe . . . 8

2.2.2 Outliers Removal . . . 8

2.2.3 NaN Handling. Data Imputation . . . 10

2.3 Data Transformation . . . 10

2.3.1 Feature Encoding . . . 10

2.3.2 Feature Scaling . . . 11

3 Results and discussion 13

4 New Improvements 15

4.1 Random Forest Model . . . 15

4.2 LSTM Neural Network . . . 15

4.2.1 Splitting the data . . . 15

4.2.2 Dealing With the Imbalanced Problem. Clustering and Undersampling. . . . 15

4.2.3 Dealing With the Imbalanced Problem. Oversampling. . . . 17

4.2.4 LSTM model . . . 18

1 Problem Description

The primary objective of this project is to develop an efficient classification model to categorize water consumers into different types. The necessary information for this task is available in a provided dataset, which encompasses a wide range of variables, including the year, month, id of the consumers, the geographical location of the installation, and the specific type of consumer.

1.1 Dataset Variables

- **Year and Month**: Allows for the analysis of seasonal patterns and trends over time.
- **Consumer Number**: Unique identifier of a given consumer.
- **Installation Zone**: Geographical information relevant for understanding variations in consumption based on location.
- **Consumer Type**: The key variable for classification, defining the category to which each consumer belongs.

1.2 Classification Objective

The main focus is to use data science and machine learning techniques to develop a predictive model that will assign each consumer to a specific category. This will facilitate an understanding of consumption patterns, identification of user profiles, and enable informed decision-making regarding water supply management.

1.3 Project Significance

Effective classification of water consumers will provide local authorities and service providers with a detailed insight into consumption distribution and behavior. This will enable the optimization of resource planning, enhance water management efficiency, and anticipate potential issues or changes in demand.

This report will concentrate on exploratory data analysis, data preparation for modeling, selection of relevant features, and the implementation of a classification model that aligns with the project objectives.

1.4 Dataset Characterization: Exploration and Initial Analysis

The dataset provided for this project spans seven years, from 2013 to 2020, excluding the year 2015. This temporal range covers a total of 49 installation zones, characterized from 1 to 49. During this period, data is observed for various variables, with a particular emphasis on the key variable of water consumption.

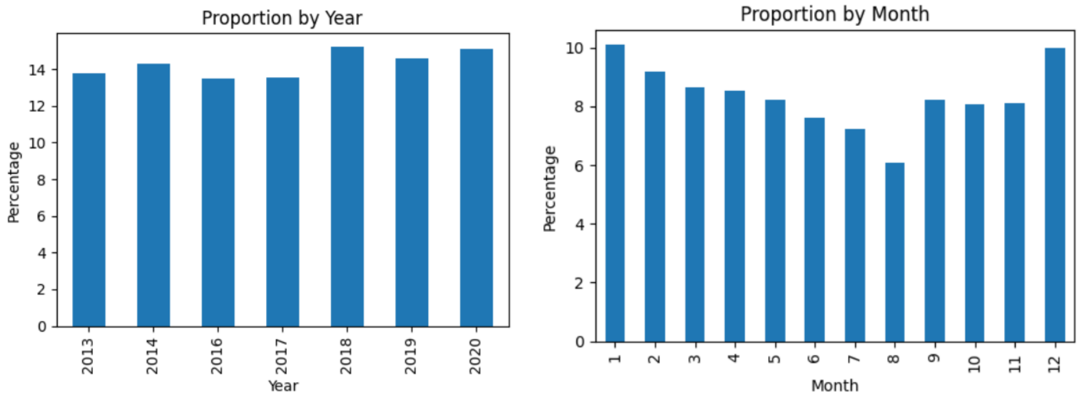
- **Temporal Details**: Included the years 2013, 2014, 2016, 2017, 2018, 2019 and 2020.
- **Installation Details**:
  - Number of installation zones: 49.
- **Consumer Types**: The dataset comprises seven different consumer types, each representing a unique category based on specific characteristics. The *rural* categories are related to agro-production consumers in varying sizes:
  - Domestic: Residential consumers characterized by typical household consumption patterns.
  - Industrial: Consumers associated with industrial activities and production.
  - Construction: Consumers linked to construction activities.
  - Low Income Families: families with social security support.
  - Rural Domestic: small/familiar agro-production companies.
  - Rural Commercial: mid-sized companies.
  - Rural Expansion: others agro-production consumers that don't categorize as one of the other categories.
- **Water Consumption Statistics**:
  - Total number of consumers (*train.csv*): 27,632.
  - Range of consumption values: From 0 to 4,978 cubic meters.
  - Number of samples with water consumption equal to zero: 61,658
- **Data Observations**: This zero values identified in the consumption column could indicate either no consumption or missing data.

2 Methodology

2.1 Exploratory Data Analysis (E.D.A)

2.1.1 Graphs and Descriptive Statistics

The analysis began by examining the distribution of the zero values over the years, revealing information crucial for subsequent data treatment. The proportions of that values are relatively similar (Figure 1), ranging approximately from 13% in 2013, 2017, and 2016 to around 15% in 2018 and 2020. It's important to note that these percentages refer to the total number of zero consumption values, not the overall sample size [1].



(a) Missing values proportion by year.

(b) Zero values proportion by month.

Figure 1: Zero consumption proportion by year and month.

Upon repeating the same analysis for each month, more pronounced differences emerge. The months with the highest percentage of zero values are January, February, and December (Figure 1). In contrast, the lowest percentages are found in June, July, and August.

When analyzing by consumer types (Figure 2), the *Rural Expansion* and *Construction* categories stand out with a high percentage of zero values. Conversely, consumer types such as *Domestic* and *Low-Income Families* exhibit lower percentages [2].

[1] Eg. out of the 61,658 existing zero values, 15.19 % corresponds to the year 2018. If we calculate the percentages relative to the total number of samples, the values are much lower, from 2.5 to 2.8 %.

[2] Note that in this case, the percentages are relative to the total number of samples for each Consumer type. Eg. out of the 890 rural expansion samples, 41,34 % (368) of them corresponds to zero consumption values.

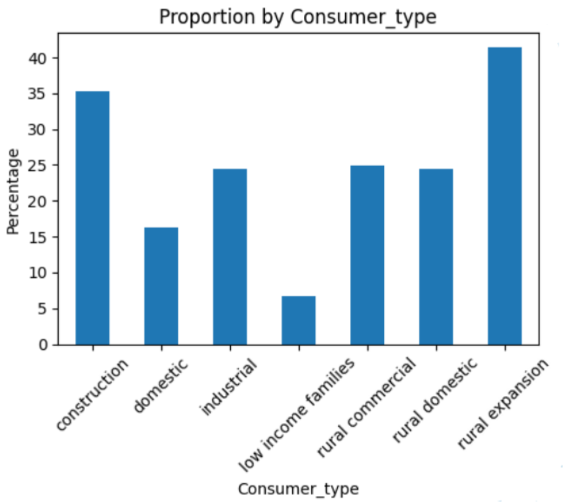


Figure 2: Proportion of 0 m3 Consumption value by Consumer$_type$.

2.1.2 Other Descriptive Graphs

We now continue with the Exploratory Data Analysis (E.D.A) process, in order to gain useful insights that could help at the processing or modelling of the problem. All of these aspects are perceptible through various graphs such as heat maps and temporal visualizations.

To explore whether different consumer types exhibit distinct monthly behaviors, we have generated plots illustrating the percentage of zero values for each month and consumer type (see Figure 3).

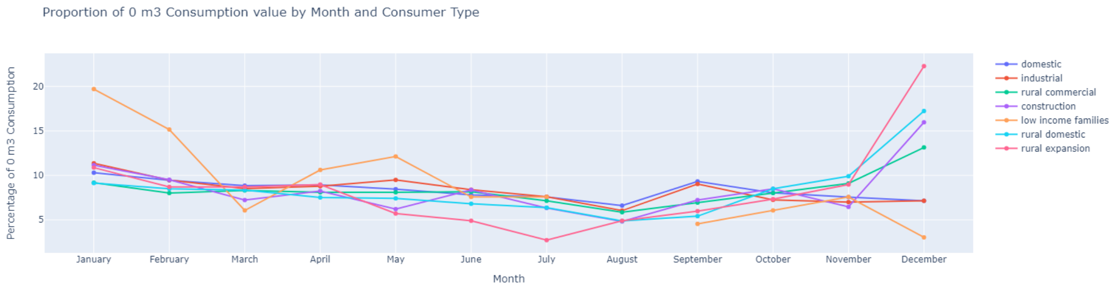


Figure 3: Proportion of zero cubic meters by month and consumer type.

Upon observing the graph, intriguing patterns emerge. The three rural categories—rural domestic, commercial, and expansion—exhibit analogous behavior, going up from September, reaching its maximum over December/January and going down until summer, July, August. Additionally, the construction category follows a similar pattern to rural categories but with a higher peak in June.

This behaviour may indicate that behind the zero values there is a mixed contribution of data loss and real zero consumption, as we can guess by observing the graph for the rural categories.

Domestic and industrial categories display comparable and more stable behavior throughout the year. Rural domestic and rural commercial also exhibit a similar trend until August, where they diverge. The low-income families category (Figure 4) displays a distinct behavior from the rest. It is possible to see this information in the following graph.

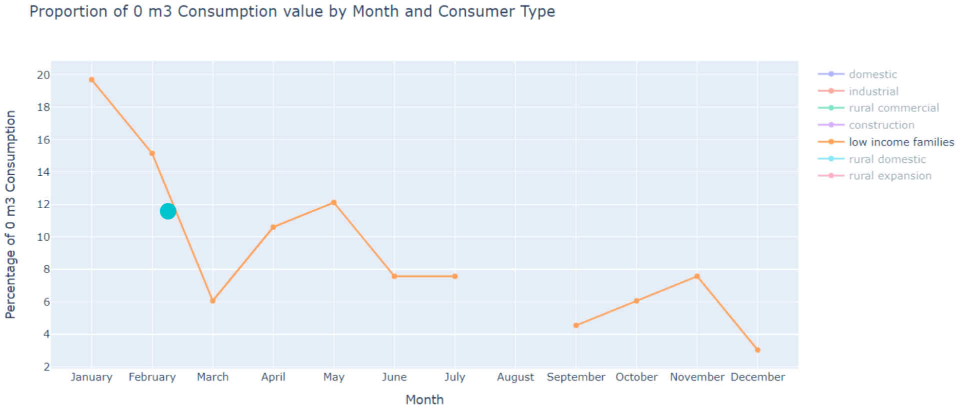


Figure 4: Proportion of zeros cubic meters for Low Income Families.

Using a basic scatter plot (Figure 5) as a visual aid, it is evident that there are some extreme values (outliers) in the data that is necessary to be addressed to enhance the model accuracy.

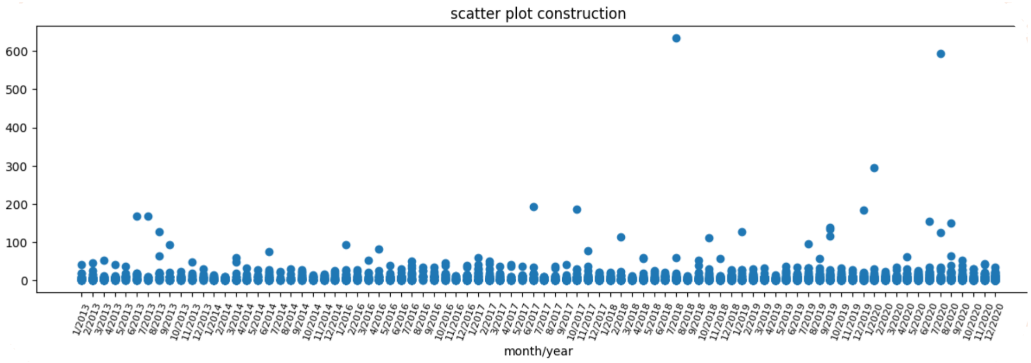


Figure 5: Scatter Plot.

2.1.3 Stats by Installation Zone

There are 49 installation zones, but certain consumer type categories are concentrated in a few of them. In fact, some zones have only one type of consumer. This feature could prove useful for the classification task. In the Figure 6 is possible to see a heatmap of consumer type by installation zone [3].

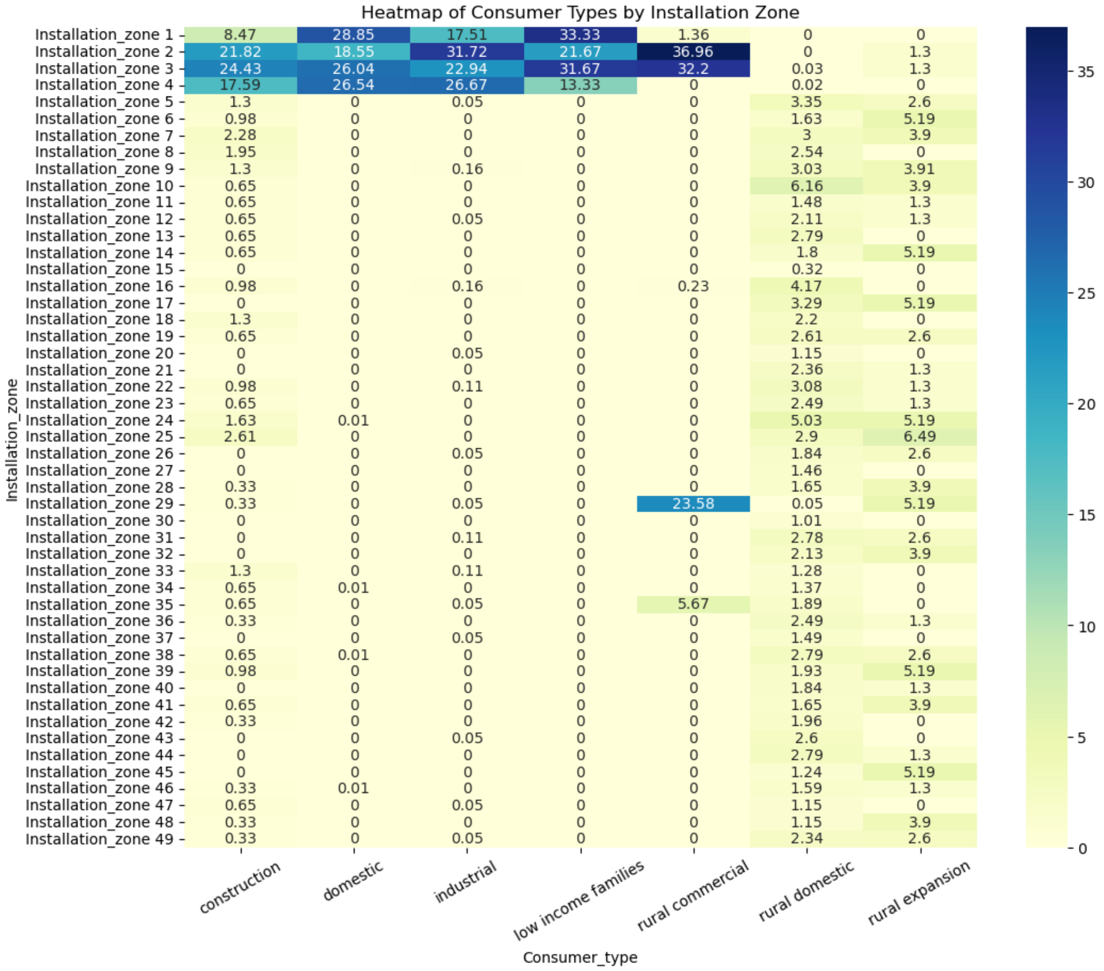


Figure 6: Heat Map of Consumer Types by Installation Zone.

Broadly, the construction, domestic, industrial, and low-income families categories are situated in Installation zones 1 to 4 (with different distributions among them). On the other hand, rural commercial, rural domestic, and rural expansion are more evenly distributed across the Installation zones, especially rural domestic and rural expansion.

[3] The values are normalized for each category.

2.2 Data Preprocessing

Includes the steps as NaN removal/imputation or handling with outliers, oriented towards the preparation of the data for the posterior phases.

2.2.1 Completing the Dataframe

When we have missing data for a particular month and consumer, we actually doesn't have the entire sample (row), so we can't just replace the NaN (because they doesn't exist). In this case, the idea is to create a complete dataframe with all the possible combinations of year, month and consumer number, and then merge it with the original dataframe. This way, we will have a complete dataset with all the months for each consumer number, and the missing values will be NaN. Then we can proceed with handling the NaN values.

For this task we used *Itertools.product()* in order to generate all possible combinations of the 3 variables (year, month and Consumer type). Now, for each consumer, we have the rows for all the months, although some of them will be NaN.

This expansion transforms the dataset from around 600,000 samples before processing to approximately 2,000,000 after that.

2.2.2 Outliers Removal

As we have seen in previous graphs, There are some outliers in the consumption values, specially for the "industrial" category. There are also some extreme values in the "domestic", "construction" and "rural domestic" categories (Fig. 7). This could represent a measurement error or a real extreme value (eg. a big company with a high water consumption, refilling a pool, etc.). In any case, it is important to note that these outliers could affect the performance of the classification model. Also, the values are not normally distributed.

There are different available methods to remove outliers. Some of them are more robust than others or assume a normal distribution of the data.

In this case, we use the IQR method, which is an easy and robust method to remove outliers. The IQR is the difference between the 75th and 25th percentiles of the data. The outliers are defined as the values that are below the 25th percentile $- k \cdot$ IQR or above the 75th percentile $+ k \cdot$ IQR. The value of k is usually 1.5, but it can be adjusted to 3 depending on the data.

The distribution of consumption values resulting after this processing step can be seen in Figure 8. Also, a scatter plot is shown after the processing step for the construction category (Fig. 9.

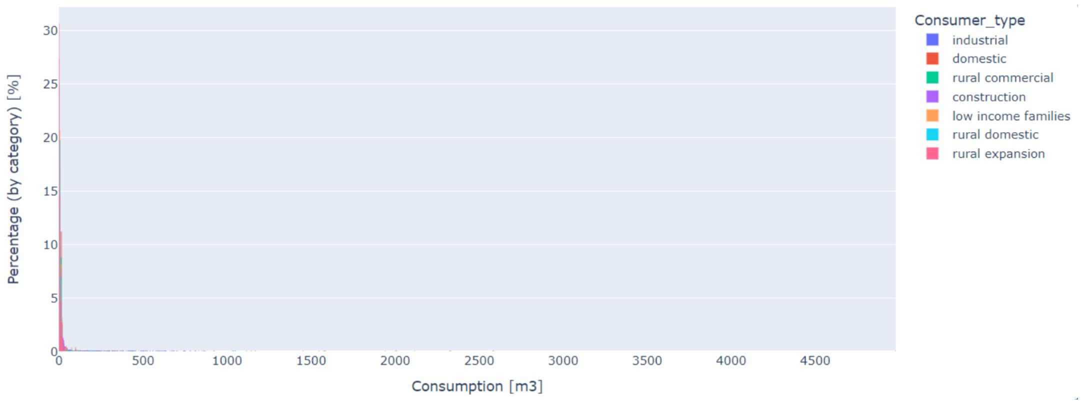


Figure 7: Consumption before removing outliers.

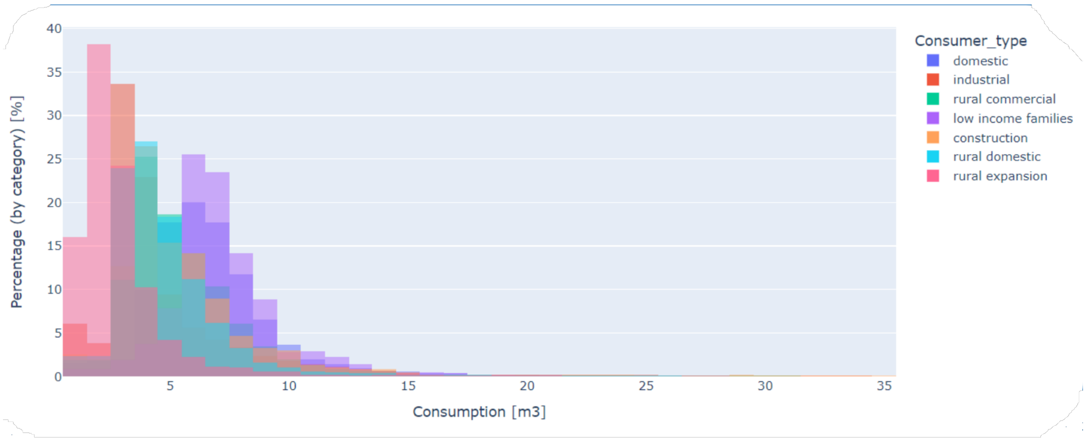


Figure 8: Consumption after removing outliers.

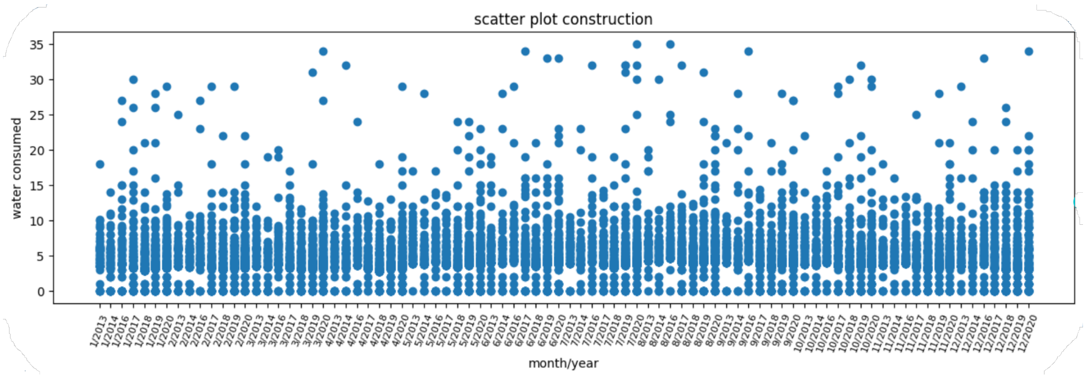


Figure 9: Scatter plot after removing outliers.

2.2.3 NaN Handling. Data Imputation

Next, we proceed with Data Completion. We have to decide what to do with the new NaN values generated by the operation of "completing" the dataframe. We decided to follow a mixed approach: First, we set a threshold so that if there are more than X months missing for a particular consumer, we proceed to remove that year. If there is only missing data for a few months, we could impute them with the mean of the mean values for that month and consumer type category that the user belongs to, and the year mean of the consumer, so that we get balanced values and we don't change too much the distribution of the data.

In this case the threshold of missing months per year was empirically set to 10, so that applying the mentioned procedure, the dataset size decreased to nearly 1,000,000 samples.

For data imputation, users with a number of missing months below that threshold are replaced with the average value of [4]:

- The mean consumption value of the particular consumer for the specific year.
- The mean consumption value of the *Consumer type* category to which the consumer belongs (for the specific year).
- The monthly mean consumption value of the *Consumer type* category to which the consumer belongs (considering all years).

2.3 Data Transformation

2.3.1 Feature Encoding

Firstly, we need to split the data into training and test sets. We will use the training set to train the model and the test set to evaluate its performance. We undertake this initial step to avoid data leakage, ensuring that parameters for subsequent steps (encoding, scaling) are trained exclusively on the training set and then applied to the test set. We have tried different approaches for the data splitting, at first place we present the results obtained with a time-based split, i.e. we use the three first years for training and the last three for testing.

The "Month" feature exhibits a cyclical nature. To account for this dependence, we transform the feature into two new ones, by the means of trigonometric functions, helping to represent the cyclic nature of this feature.

[4] As we are going to compute the mean values for each consumer and year, it is better to remove the outliers first.

The *Consumer type* and *Installation zone* features are categorical. To use them in the classification task, encoding is necessary. There are different methods for this, outlined below:

- One-hot encoding creates a new column for each category, with a 1 if the sample belongs to that category and 0 otherwise. However, this method is not recommended for high cardinality features, such as in this case where we have 7 categories for *Consumer type* and 49 for *Installation zone*.
- Label encoding: it assigns a number to each category, potentially implying a ranking. It is not recommended in this case, similar to one-hot encoding.
- Frequency encoding assigns a number to each category based on its frequency in the dataset.
- Hashing encoding, similar to one-hot encoding, uses a hashing function to reduce the number of features, making it useful for high cardinality features.
- Finally, and the chosen method, is target encoding. Encodes the categories by replacing them for a measurement of the effect they might have on the target. In this case this is achieved by computing the posterior probability of the target given particular categorical value and the prior probability of the target over all the training data.

2.3.2 Feature Scaling

Necessary for some algorithms that are sensitive to the scale of the features or distance based, like SVM, KNN, Neural Netwworks, etc. The principal methods include standardization, min-max scaling (normalization), robust scaling and power transformation.

Standardization

Standardization, a widely employed scaling method, adjusts the features to have a mean of 0 and a standard deviation of 1. This transformation is beneficial for features that adhere to a normal distribution but is sensitive to the presence of outliers. The computation of standardization is expressed as follows:

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

Min-max scaling (Normalization)

Min-max scaling adjusts data values to fit within the range of 0 to 1, making it particularly suitable for features with varying scales. However, this method can be influenced by outliers. The calculation involves:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Robust scaling

Robust scaling was the selected method, and it shares similarities with standardization; however, instead of dividing by the standard deviation, it employs the interquartile range (IQR) to scale the data. This method proves advantageous for features containing outliers, as it exhibits greater robustness in handling them.

$$x_{scaled} = \frac{x - \mu}{Q_3 - Q_1}$$

Power transformation

Implement a power transformation on the data to enhance its Gaussian-like characteristics. Subsequently, standardize the transformed data for further analysis and modeling. This approach aids in addressing non-normality in the original distribution.

Below (Fig. 10) it is shown a time series graph of the consumption by consumer type, scaled and without outliers [5].

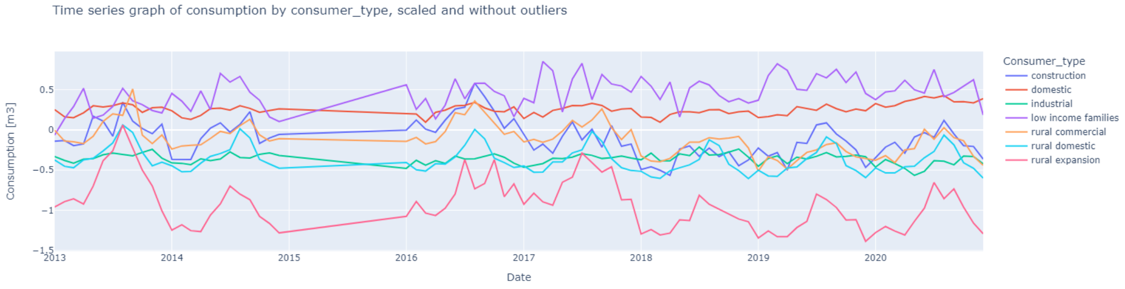
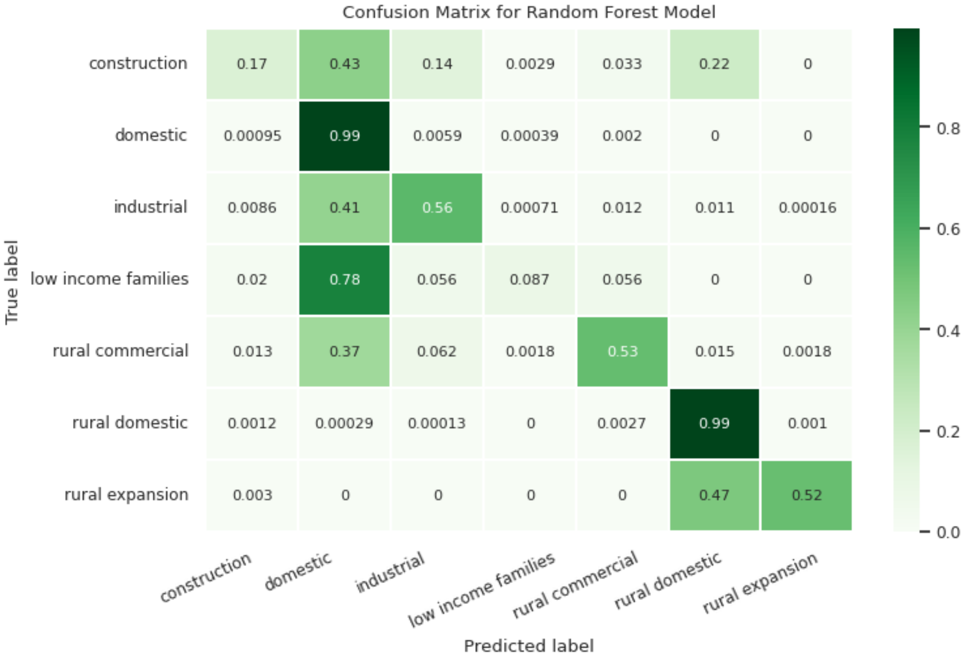


Figure 10: Time series scaled.

[5] The line shown in the figure at year 2015 is a graphical artifact. In fact we don't have data for this year and we decided not to generate artificial one.

3 Results and discussion

In this stage of the project, a Random Forest model was chosen for classification, trained on a specific dataset, and subsequently evaluated using the time-based split alredy mentioned. The generated confusion matrix provides a detailed view of the model's performance. This matrix is shown in the Figure 11.



	construction	domestic	industrial	low income families	rural commercial	rural domestic	rural expansion
construction	0.17	0.43	0.14	0.0029	0.033	0.22	0
domestic	0.00095	0.99	0.0059	0.00039	0.002	0	0
industrial	0.0086	0.41	0.56	0.00071	0.012	0.011	0.00016
low income families	0.02	0.78	0.056	0.087	0.056	0	0
rural commercial	0.013	0.37	0.062	0.0018	0.53	0.015	0.0018
rural domestic	0.0012	0.00029	0.00013	0	0.0027	0.99	0.001
rural expansion	0.003	0	0	0	0	0.47	0.52

Figure 11: Confusion Matrix for Random Forest Model.

The obtained results are as follows:

- Training Score: 0.977
- Test Score: 0.952
- Precision score: 0.712
- Recall Score: 0.551
- Accuracy score: 0.952
- F1 Score: 0.609

While the high training score suggests effective learning of patterns in the training data, the evaluation with unseen data reveals areas for improvement (overfitting). Despite a robust test score (accuracy) of 0.952, precision of 0.712, recall score of

0.551, and F1 score of 0.609 indicate aspects that could be optimized. In such an imbalanced dataset it is very important to notice that the accuracy is not a good metric to assess the performance of a model, as it is obtained as the number of correct samples out of the complete dataset. So a model that always predicts the majority class will achieve a very high accuracy.

A precision of 71.2% highlights the model's ability to correctly classify positive instances among all it has identified as such. This implies that when the model predicts an instance as positive, there's a 71.2% chance that this prediction is correct. On the other hand, a recall of 55.1% indicates that there is room for improvement in the model's ability to detect all real positive instances. This means that out of all the actual positive cases present, the model was only able to correctly identify 55.1% of them.

In conclusion, the Random Forest model has demonstrated solid performance in classifying water consumers, exhibiting good generalization capabilities. However, a more detailed evaluation is needed to improve the balance between precision and recall. This additional analysis can contribute to significant enhancements in accurately identifying different types of consumers, thereby increasing the effectiveness and utility of the model.

4 New Improvements

Below, we will list some of the improvements and changes made to try to improve the classification model.

4.1 Random Forest Model

First, for the random Forest model, we tried two different approaches to get the final prediction for a consumer:

- Compute the mode of the prediction for each "Consumer number" and assign it to the consumer.
- Among all of the samples prediction for a particular consumer, assign the less frequent predicted class. It is not a common approach, but in such a extremely unbalanced dataset, it could be useful to avoid the model to predict always the majority classes. For example, for a consumer type "low income families", it is very unlikely that the model predicts the 50% or more of the samples as "low income families". So, if at least one of the samples is classified as that, we proceed to assign it to the consumer.

4.2 LSTM Neural Network

We also tried a new and different approach, which is to treat the problem as a time series (see time_series.ipynb). To do this we first have to go through a series of steps that will help us improve the classification metrics. Also it was helpful to make a file to store some of the new custom helper functions (file functions.py)

4.2.1 Splitting the data

First, we tested an alternative to the time splitting, and developed a function, `split_by_id()` for splitting the dataset according to id. That is, there are different consumers in the training and test dataset, and they are not repeated. This function also maintains the proportion of the target variable (Fig. 12). It takes into account the Consumer type variable, so that the proportion of each Consumer type is the same in both sets.

4.2.2 Dealing With the Imbalanced Problem. Clustering and Undersampling.

At first place, the idea is to reformat the data in a way that each row (sample) corresponds to a unique consumer, and each column corresponds to a unique combination of month and year (date). Then, we can apply clustering and undersampling

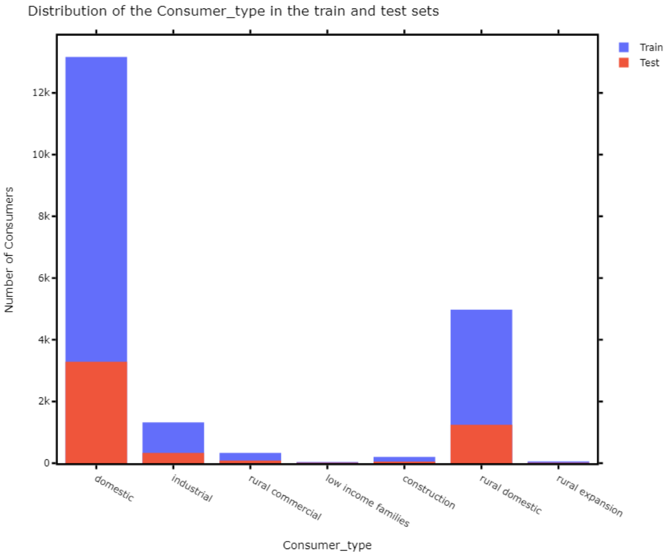
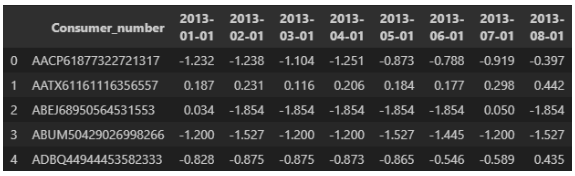


Figure 12: Distribution of consumers in training and testing datasets.

techniques to the overrepresented classes, in order to reduce the dimensionality of the dataset and to find patterns in the data. This could be useful to find groups of consumers that have similar consumption patterns, so that we can reduce the dimensionality of the dataset but also mantain the information of the original features.

Changing the shape of the dataset is achieved with the pandas function `pivot_table`, resulting in the following example:



	Consumer_number	2013-01-01	2013-02-01	2013-03-01	2013-04-01	2013-05-01	2013-06-01	2013-07-01	2013-08-01
0	AACP61877322721317	-1.232	-1.238	-1.104	-1.251	-0.873	-0.788	-0.919	-0.397
1	AATX61161116356557	0.187	0.231	0.116	0.206	0.184	0.177	0.298	0.442
2	ABEJ68950564531553	0.034	-1.854	-1.854	-1.854	-1.854	-1.854	0.050	-1.854
3	ABUM50429026998266	-1.200	-1.527	-1.200	-1.200	-1.527	-1.445	-1.200	-1.527
4	ADBQ44944453582333	-0.828	-0.875	-0.875	-0.873	-0.865	-0.546	-0.589	0.435

Figure 13: Enter Caption

Next, the idea is to apply undersampling by finding clusters in the data and then selecting representative samples from each cluster. To perform this task we had to create the class `OptimalClusterFinder`, which, as its name suggests, is passed the data for a consumer type, and returns the optimal number of clusters to be used in the next function. On the other hand, the function `subsample_dataset()`, first applies PCA to reduce the dimensionality of the data before finding the clusters, then finds the specified number of clusters, and finally, randomly chooses a sample from each of those clusters until the specified sample size is reached.

4.2.3 Dealing With the Imbalanced Problem. Oversampling.

The following graph (Fig. 14) shows the temporal consumption for the less represented Consumer types (low income families, construction, rural expansion and rural commercial). It displays the average group consumption with a shaded area that represents the interquartile range (25-75 %), in order to diminish the effect of outliers. It shows that in theory, is possible to do oversampling for these groups, by sampling from its statistical distribution.

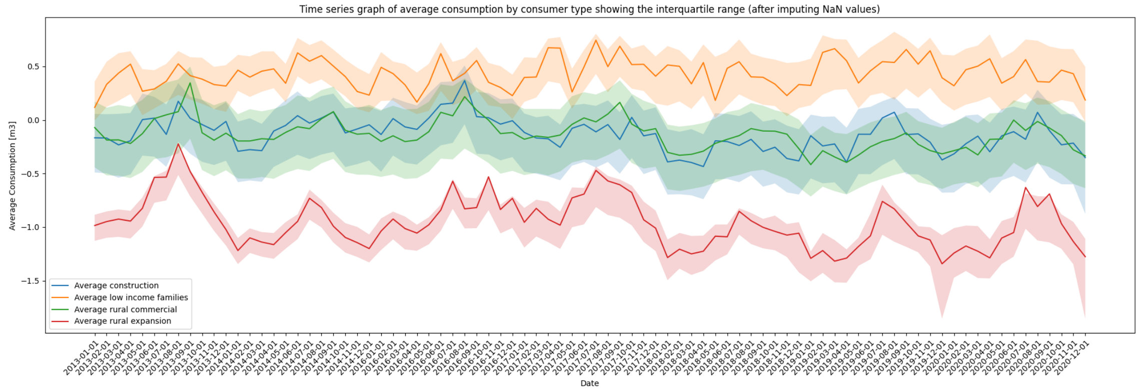


Figure 14: Time series graph of average consumption by consumer type.

For that job, we use the custom function `generate_synthetic_samples()` to create new time series data k deviations around the mean for each underrepresented class.

The following graph shows some of the new generated samples along with the group mean and deviation.

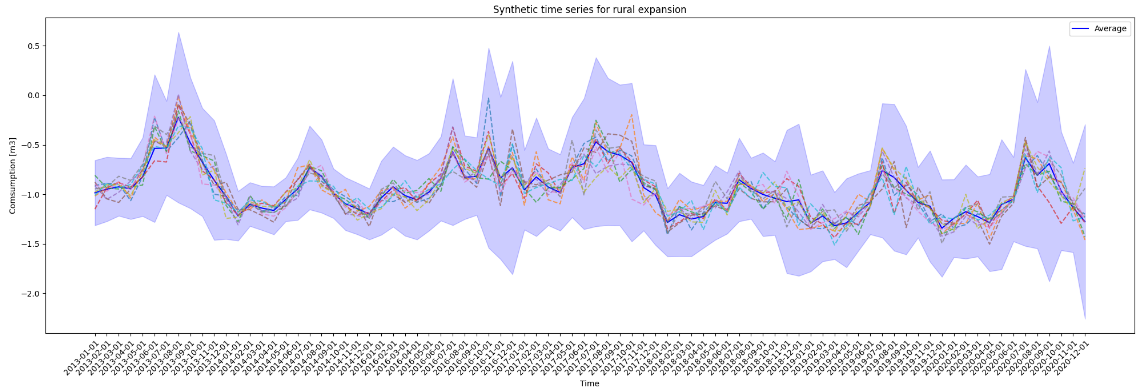


Figure 15: Synthetic time series data for rural expansion class.

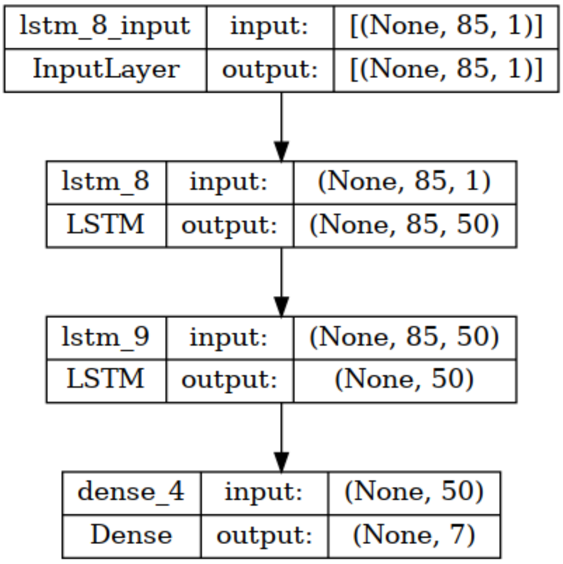


Figure 16: LSTM layers and units.

4.2.4 LSTM model

Finally, we train the LSTM model taking into account the input format it requires, and logging the training and test metrics with the help of the Weights & Biases platform, obtaining suspiciously high metrics in both training and validation, of:

- Training Precision: 0.95
- Training Recall: 0.94
- Validation Precision: 0.94
- Validation Recall: 0.91