PREDICTING (MISSING) ENERGY LABELS

A COMPARATIVE STUDY ON RANDOM FOREST, XGBOOST AND TABNET

ILSE D'HOOGHE

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

728541

COMMITTEE

dr. Görkem Saygili
dr. Nevena Ranković

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

May 19th, 2023

WORD COUNT

8799

ACKNOWLEDGMENTS

I would like to thank my supervisor, Gorkem Saygili, for his guidance, support and kindness during the process of writing my master's thesis. The guidance sessions were always helpful, and he also managed to put some humour, and laughs into the sessions.
I would also like to thank The Green Land for facilitating in finding this project, and for the mental support. In addition, I would like to thank my external supervisor Stephan Preeker (VNG) for his knowledge, and support.
Finally, I would like to thank my dear mother, brother and other family relatives for their support and confidence in me.

CONTENTS

PREDICTING (MISSING) ENERGY LABELS

A COMPARATIVE STUDY ON RANDOM FOREST, XGBOOST AND TABNET

ILSE D'HOOGHE

Abstract

Among municipalities in the Netherlands, numerous energy labels of houses remain unknown. This complicates the process of transitioning houses in specific neighbourhoods to renewable heating methods, in order to reduce the Netherlands' gas emissions. Lower energy labels in a neighbourhood indicate a greater urgency to undertake this transition. This study investigated the possibility of predicting (missing) energy labels of houses in Brabant using Random Forest, XGBoost, and TabNet. Compared to other studies, this study contained aggregated consumption data, and lacked data on the building envelope.

The impact of the class imbalance was mitigated by the utilisation of SMOTE-Tomek, resulting in a substantial improvement in predicting the minority classes E, F and G. Unfortunately, the classifiers continued to perform poorly, even after utilizing a large proportion of unlabelled data. RF, XGBoost, and TabNet achieved F1-scores of 0.478, 0.483, and 0.405, respectively. The poor performances were presumably due to aggregated data, and the lack of data on the building envelope, resulting in confusion among classes. Future research is therefore encouraged to include such data.

1 INTRODUCTION / PROBLEM STATEMENT / RESEARCH GOALS

The transition to sustainable heating methods in houses plays a crucial role in mitigating climate change, and reducing gas emissions. Energy labels serve as valuable indicators in identifying neighbourhoods in need of a transition. However, the missing energy labels for certain houses poses a challenge for Dutch municipalities. Therefore, this thesis aims to investigate the usability of the data to predict energy labels.

The temperature of the Earth has been rising due to the increasing levels of green house gases (Lindsey & Dahlman, 2023). Hence, the Paris

Agreement's objective to limit global warming and mitigate climate change (United Nations, 2019). Many parties, including the Netherlands, agreed to reduce gas emissions by 95% by 2050 (Dutch Government, 2019). Given that the Netherlands relies mainly on natural gas to heat houses, the country is required to transition to renewable heating methods. Under the Paris Agreement, the Dutch municipalities will develop a Transition Vision on Heat every five years, outlining the order and timeline for transitioning neighbourhoods away from natural gas (VNG, 2020).

The Association of Netherlands Municipalities[1] (VNG) has an important role in supporting municipalities, and therefore developed the Energy Transition Built Environment[2] (DEGO) to assist municipalities in their decision-making regarding the Transition Vision on Heat. The DEGO displays a map of buildings in the Netherlands, including details such as the construction year, energy label, gas, and energy consumption.

Energy labels provide information about the energy efficiency of a building (Rijksoverheid, 2014). The labels are ranked from best (A++++) to worst (G), and intended as incentives to encourage energy-saving measures, such as the installation of solar panels or renewable heating systems. The amount of low energy labels (D-G) in a neighbourhood could be used by municipalities as a measure to prioritise its transition.

Figure 1: Missing Energy Labels (grey surfaces) in the DEGO. Source: DEGO.vng.nl.

However, as shown in Figure 1, the DEGO lacks data on the energy labels of numerous houses. This is due to the fact that a house has never been sold, rented or rebuilt since the time energy labels became mandatory in 2008 (Rijksoverheid, 2016). Although the map becomes complete over time, it remains inconvenient for officials to base decisions on a map that

[1] Dutch: Vereniging Nederlandse Gemeenten
[2] Dutch: Datavoorziening Energietransitie Gebouwde Omgeving, https://dego.vng.nl

lacks crucial information. Therefore, this study aims to examine if Machine Learning algorithms are able to predict the (missing) energy labels of houses with the available data in the DEGO.

There are several challenges to consider. Firstly, limited research has been conducted on predicting energy labels of houses (Cai et al., 2019; Tsoka et al., 2021). Secondly, the dataset used contains a considerable amount of unlabelled data (approximately 51%). Lastly, there are severe class imbalances, which may distort the prediction model (Chawla et al., 2002). In this study, it is considered more important to predict the lower-ranked energy classes, as these houses require a transition.

1.1 *Research Questions*

Based on the aforementioned challenges, an overarching research question was formulated:

To what extent can Machine Learning algorithms predict the energy labels of houses in the province Noord-Brabant of the Netherlands?

To answer the broad research question, three sub-questions were formulated:

RQ 1 *How do Machine Learning algorithms (Random Forest & XGBoost), and Deep Learning algorithm TabNet perform in predicting the energy labels of houses with imbalanced data?*

RQ 2 *Can SMOTE-Tomek overcome the class imbalance, and improve the performances of the classifiers?*

RQ 3 *Can Semi-supervised Learning improve the prediction performance of the best-performing model via exploiting unlabelled samples in the dataset?*

1.2 *Motivation & Relevance*

This research is important from both social and scientific perspectives. It supports municipalities in their process to transition homes to renewable energy, and it provides them with insights into the usability of the data in the DEGO for data science methods. Scientifically, it contributes to a comprehensive study aimed at predicting energy labels using house characteristics, and aggregated consumption data. Furthermore, it addresses challenges such as class imbalance, and missing labels by utilizing SMOTE-Tomek, and semi-supervised learning (SSL) to improve predictions. This

study is of relevance due to the limited research in this field, and the use of a novel dataset, and algorithm (TabNet).

2 ETHICS / DATA / TECHNOLOGY STATEMENT

The dataset used in this study originates from the DEGO, which is maintained by the VNG, and was obtained with consent. The dataset contains data about buildings in Noord-Brabant. In the interest of privacy of residents, data from grid operators have been aggregated. Moreover, information on human subjects is not included. The code[3] is available via GitHub[4].

ChatGPT has been used as a debugging tool to resolve coding errors (OpenAI, 2021). For assistance in academic writing and grammar, the author used Thesaurus [5] (Thesaurus, 2023) and LanguageTool (Naber & Miłkowski, 2005).

The figures, except for Figures 1, and 6 (which were obtained with consent), were made by the author.

[3] Inspirational code snippets from, for instance, Stack Overflow and Kaggle have been referenced in the notebooks.

[4] https://github.com/Ilsedh/thesis-dss-ilse-dhooghe

[5] https://www.thesaurus.com/

3 RELATED WORK

3.1 *Predicting Energy Consumption, and Energy Labels*

Recent studies mainly focused on utilising Machine Learning to forecast energy consumption to help achieve climate goals (Fathi et al., 2020; Olu-Ajayi et al., 2022; Papadopoulos et al., 2018; Seyedzadeh et al., 2019). Predicting energy consumption involves predicting a continuous value rather than predicting a class. Nonetheless, insights from energy consumption studies are valuable for this thesis, as similar data characteristics are used.

The most widely used algorithms for estimating energy consumption are Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF), and Artificial Neural Networks (ANNs) (Olu-Ajayi et al., 2022; Walker et al., 2020). According to Walker et al. (2020), the RF algorithm is a more effective algorithm than a single Decision Tree, as it stacks single Decision Trees. The principle of stacking different weak learners is referred to as an ensemble algorithm. Ensemble algorithms help to tackle the problem of overfitting [6]. Nowadays, the ensemble algorithm Extreme Gradient Boosting (XGBoost), is even of greater popularity and used in several studies (Chen & Guestrin, 2016b; Ding et al., 2021; González-Briones et al., 2019; Goyal et al., 2020; Seyrfar et al., 2021). Seyrfar et al. (2021) compared XGBoost's performance with that of an RF and a Neural Network in estimating the energy consumption of dwellings. The results showed that XGBoost achieved the highest accuracy rate (68%). Tree-based ensemble algorithms are highly recommended when encountering classification problems with tabular data (Shwartz-Ziv & Armon, 2022).

Besides tree-based models, Neural networks have also proven to be successful. Especially deep neural networks have been exceedingly successful in studies with image, audio, or text data (Arik & Pfister, 2021). Arik and Pfister (2021) argued that Deep Learning for tabular data is still understudied. Therefore, they proposed a Deep Learning algorithm (TabNet) that leverages sequential attention that selects the most important features at each step, which enables interpretability. Despite TabNet being a new and successful algorithm, it has not yet been used in the context of predicting energy labels.

A recent noteworthy study compared XGBoost and deep learning algorithms for tabular datasets. The study demonstrated that XGBoost

[6] Overfitting arises when a predictive model learns the training data in too much detail. This means that the model memorizes the patterns, including the noise, in the training data instead of effectively learning from the underlying patterns. This leads to poor performance on new unseen data.

outperformed TabNet on several tabular datasets (Shwartz-Ziv & Armon, 2022).

3.1.1 *Predicting Energy Ratings and Labels*

Cai et al. (2019) focused on classifying electricity consumption ratings. The authors compared SVMs, Gradient Boosting Decision Tree (GBDT), and a Back Propagation Neural Network (BPNN). The GBDT, and the BPNN had an accuracy rate of approximately 75%. The SVM slightly outperformed both models with an accuracy of around 78%, however, it had a longer CPU runtime due to the complexity of the model. The data they used in their paper was high-dimensional, which is why they used multiple feature engineering algorithms, such as RF's *feature_importances_* attribute, which will also be used in this thesis.

In the past five years, only two contiguous studies, with the same authors, aimed at predicting energy labels of residential buildings in Lombardy in Italy (Tsoka et al., 2021, 2022). The studies classified buildings according to their energy label, with the use of ANNs. They treated the task as a multi-class classification problem, and discarded the order between energy labels.

They mainly focused on explainable AI methods, rather than comparing different Machine Learning and Deep Learning algorithms in predicting energy labels. The ANN from the 2021 study had an accuracy score of 69.93% on the test set. Their dataset contained a total of 207,325 samples.

3.1.2 *Input Data*

The input features that were entered to the classifiers differ slightly from one study to another. Olu-Ajayi et al. (2022) stated that predicting energy consumption is complex, given that many factors are of influence, such as the consumption of the occupants, the building characteristics and physical properties. Tsoka et al. (2021) used similar features, such as the building envelope and the climate of the building, to predict the energy labels. The building envelope features included factors such as, U-values of the roof, walls, windows and basement. Glazed or opaque surfaces were also included in their study. (Tsoka et al., 2021) identified trough Explainable AI methods that the U-values for walls, and the opaque surface features had the strongest influence on the classification of energy labels.

The available data in this study was restricted. For instance, there were no characteristics of the building envelope, nor data on indoor conditions. Moreover, the information from the Dutch grid operators was aggregated, which likely resulted in the loss of valuable information at the individual

home level (Pollet et al., 2015). The only data available at individual house level were the height, area, volume and year of construction of a house.

3.2 *Dealing with Class Imbalance*

Class imbalances can lead to biased classifiers that favour the classes with more samples. According to Chawla et al. (2002), a classifier could have a strategy to simply always predict the majority class as the outcome, which can yield incorrect, misleadingly high accuracy scores. In the studies of Tsoka et al. (2021, 2022), the majority of the data consisted of class G (52.03%). Despite that they argued that input data for neural networks should be reliable and free of faulty data, they did not address which methods might have been appropriate for dealing with class imbalance. Chawla et al. (2002) created the Synthetic Minority Over-sampling TEchnique (SMOTE) which generates synthetic samples for the minority classes. Sasada et al. (2020) explained that although SMOTE can overcome class imbalance, it also causes additional complexity to the dataset, such as noisiness and overlapping patterns. This is also expected in this work, considering the many classes. SMOTE-Tomek, and SMOTE-ENN are two algorithms that combine over- and undersampling. The former aims at reducing overlapping samples between classes, and the latter tries to remove noisy samples. A disadvantage of the removal of these samples is the loss of potential important information (Yang & Li, 2022).

There are more than 80 variants of the SMOTE algorithm (Kovács, 2019). SMOTE-Tomek, and SMOTE-ENN were used in different applications. Cai et al. (2019) combined their SVM classifier with the SMOTE-ENN algorithm, which resulted in a major improvement in accuracy in predicting energy consumption of buildings. It outperformed all the other classifiers, with an accuracy rate of approximately 97%. Another study compared Borderline-SMOTE, ADASYN, SMOTE-ENN, and SMOTE-Tomek to overcome the class imbalance in bankruptcy detection (Le et al., 2018). SMOTE-Tomek, and SMOTE-ENN were the best oversampling techniques with improvements of 1.7%, and 1.8% respectively for their RF classifier.

3.3 *Utilising Unlabelled Data*

When unlabelled data is not utilised, the chances are that a considerable amount of useful information will be lost. A well-known potential solution to this is Semi-Supervised Learning (SSL). Various types of SSL algorithms have been introduced, each relying on different assumptions (Van Engelen & Hoos, 2019).

One of these algorithms is Self-Training, and was proposed by Yarowsky (Yarowsky, 1995). The algorithm has been widely used in computational linguistics, and was composed of two loops. The first loop trained a traditional supervised classifier to map each input to a corresponding label. The second loop utilised the classifier's function to map the unlabelled inputs to their corresponding labels (Abney, 2004). Self-Training assumes that the labelled data are representative for the entire dataset, ensuring that the classifier's generated predictions are trustworthy (Yarowsky, 1995).

A few years later, Zhu and Ghahramani (2002) proposed the Label Propagation Algorithm. This approach differs in the respect that it is a graph-based algorithm. This approach relies on local similarity, meaning that if samples in the data are similar, then those samples are likely to have the same label (Ligthart et al., 2021; Van Engelen & Hoos, 2019).

SSL techniques have demonstrated to be effective in solving the problem of limited availability of labelled data (Barreto et al., 2020; Ligthart et al., 2021). It is noteworthy that SSL can also instead decrease performance (Van Engelen & Hoos, 2019).

3.4 *Research Gap*

The literature shows that sufficient research has been done on predicting energy consumption. However, there is a lack of research on predicting energy labels, particularly when dealing with class imbalance and a substantial amount of unlabelled data. Additionally, the working of the TabNet algorithm on predicting energy labels has not been studied yet.

Moreover, the dataset used in this study contains a few aggregated features, and lacks information on the building envelope, as discussed in Section 3.1.2. This study investigates whether the available data in the DEGO is sufficient for accurately predicting energy labels. Although insufficiency could lead to a less performing model, it is an opportunity to examine data that has never been scrutinized like so before.

4 METHOD

This section outlines the methodology used to obtain answers to the research questions. The dataset, and the various methods used to analyse, clean, and preprocess the data are explained. Furthermore, the algorithms and techniques used for the prediction task and how they were subsequently evaluated will be discussed.

Overview Methodology
The methodology (Figure 2) involved several steps. Section 4.2 presents the exploratory data analysis, and Section 4.3 will subsequently discuss the data cleaning and preprocessing steps. Due to missing labels, the dataset was split into labelled and unlabelled sets for future use.

For the purposes of training the classifiers (RF, XGBoost, TabNet), conducting the experiments, and subsequently evaluating them, the dataset was split into a training and a test set. The training set comprised 80% of the data. The remaining 20% of the data was used for the test set.

The algorithms used, and the experiments conducted, are explained in more depth in Section 4.4, and 4.5 respectively.

To establish baselines for evaluation, the default untuned algorithms were used (meaning the algorithms with their default settings). Evaluation metrics are discussed in Section 4.6.

The list of the software, and packages used, is provided in Appendix D, page 54.

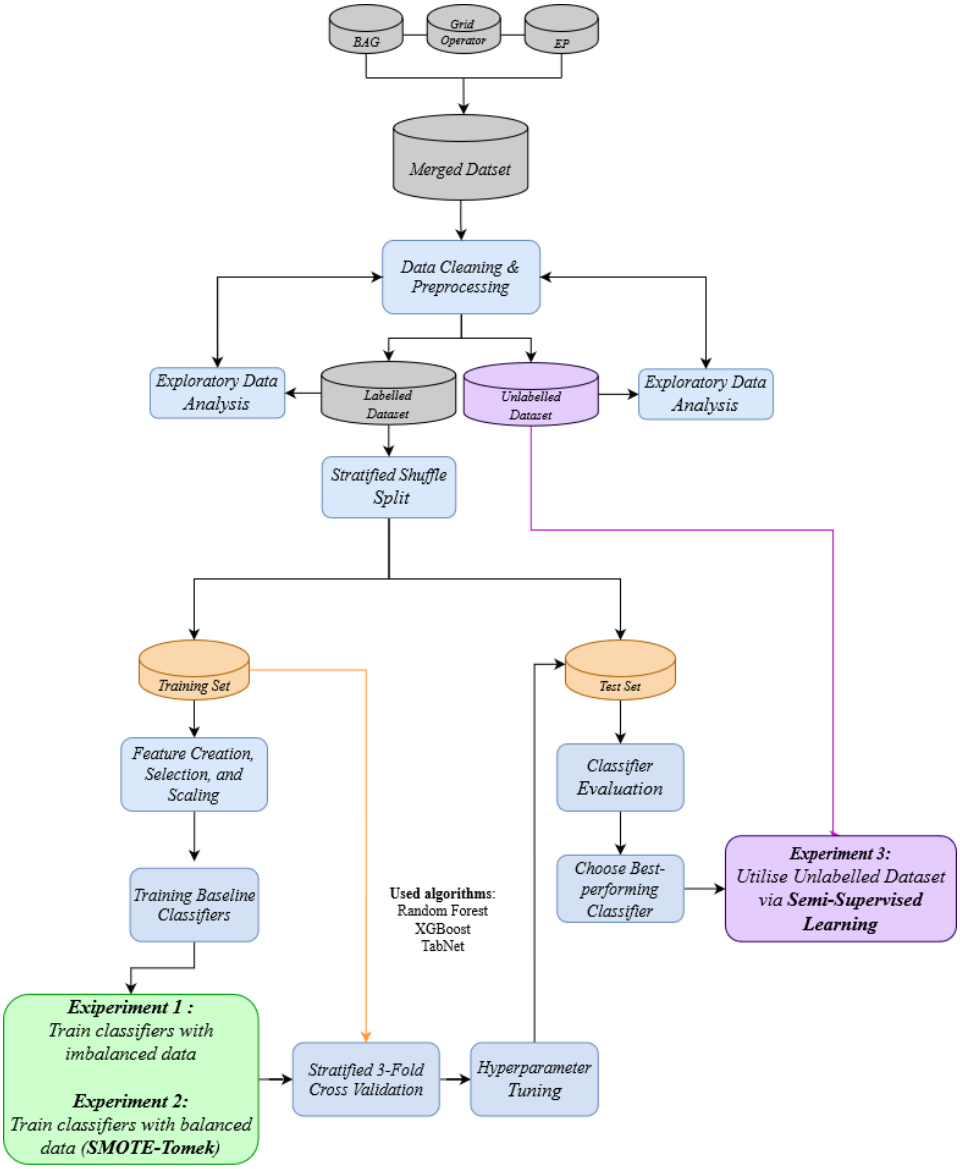

Figure 2: Overview Methodology

4.1 *Dataset Description*

The dataset used in this study contained information on buildings (offices, hospitals, stores, houses, et cetera) in Noord-Brabant in 2022. Several data sources have been merged to create this dataset. First, the BAG-register of The Netherlands' Land Registry was used. The Registry records spatial and administrative data on buildings, such as the post codes of buildings, the function of the building, the construction year, the area, and height. Secondly, the energy labels were gathered from the EP-online database from the Netherlands Enterprise Agency [7]. Lastly, small consumption data from websites of Dutch grid operators, such as Enexis, were added. Features such as annual average gas, and energy consumption were included. To ensure the privacy of the residents, the data of the grid operators were aggregated. For each observation, at least 10 connections (buildings) were aggregated.

The raw dataset contains 2,152,018 samples, and 21 features which are described in Appendix A (page 46).

[7] Dutch: Rijksdienst voor Ondernemend Nederland

4.2 *Exploratory Data Analysis*

Exploratory Data Analysis (EDA) is used by data scientists to understand and visualise the dataset at hand. In this study, the data features were examined by observing distributions, box plots, the correlation matrix, and heatmaps.

Box Plots and Distributions: A more in depth analysis of the box plots, and distributions for certain features can be found in Appendix B (47). In summary, the gas- and energy features contained a considerable amount of outliers. Additionally, the distributions for the *Area*, *Net Metering*, and *Construction Year* features were skewed.
Correlations: The correlation matrix in Figure 3 reveals that the features were not highly correlated with the target feature (the energy label). The sole feature that was moderately strong correlated (0.65) with the target label is the *Construction Year*. This indicated that newer houses tend to have better energy labels compared to older houses.

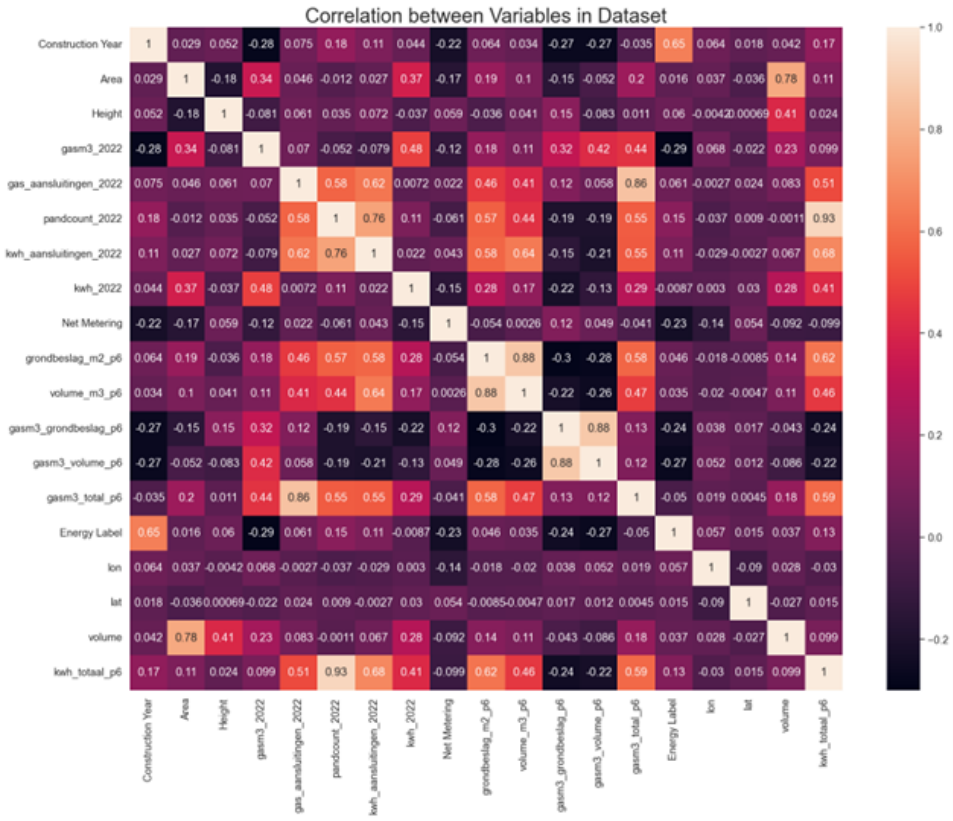

Figure 3: Correlation Matrix

In addition, as depicted in figure 4, and supported by the correlation score (-0.29), there was a negative correlation between a house's energy label and *gas consumption*. Although the correlation is moderate, it pointed out that as the gas consumption of a house increases, the energy label becomes lower. This did not hold for the *energy consumption*, which was also supported by the weak correlation coefficient (-0.0087).

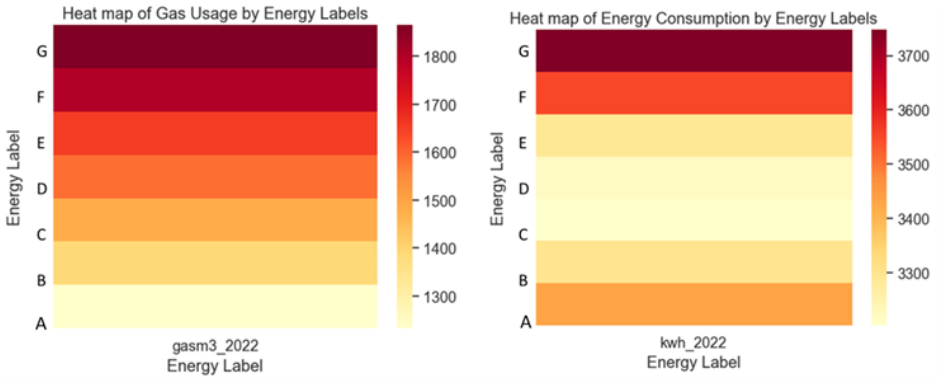

Figure 4: Heat Maps of Average Gas and Energy usage per Energy Label

Energy Label Distribution: Figure 5 shows that there was a severe class imbalance in the target feature. There were very few E, G, F, and A+ labels. It was decided to merge the different A+ classes in order to create a broader and larger A class.

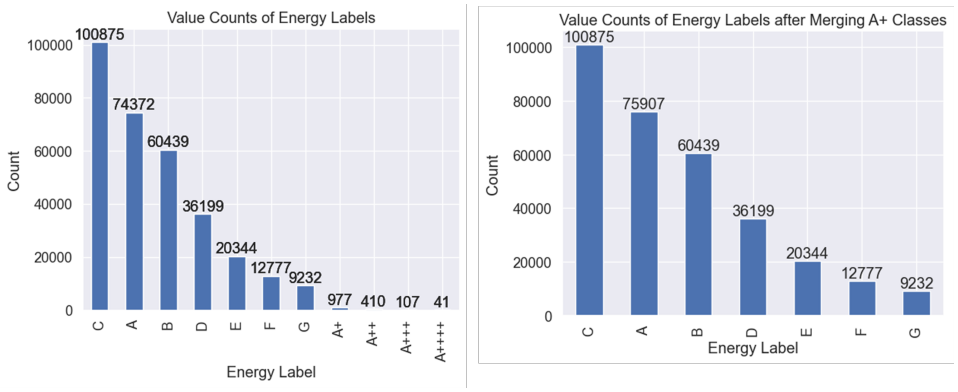

Figure 5: Distribution of the Class Labels before, and after Merging

4.3 *Data Cleaning and Preprocessing*

Data cleaning is crucial as it prevents missing values, incorrect values, and outliers [8] from entering the classifier. Clean data result in higher data quality, which subsequently improves the classifiers' performance. Moreover, data preprocessing ensures that the data have the right format and scale, and that only the most informative features get selected before training the classifier.

4.3.1 *Data Cleaning: Filtering*

This thesis focuses solely on houses in Noord-Brabant. Including other buildings, such as schools, greenhouses, and flats could have led to more complexity, as these type of buildings have varied usage patterns, which could lead to a less performing one-size-fits-all classifier (Pérez-Lombard et al., 2008).

Although the dataset was filtered to contain houses, it was not guaranteed that it contained truly only those. This was due to errors in the BAG registration of residential functions of buildings. This registration is called "function_score" in the dataset, and it indicates what proportion of a building consists of a residential function. For instance, 0% indicates no residential function, and 100% indicates solely residential function. A common error is to register a house with an attached farm or greenhouse as 100 % residential, despite the non-residential function of the attached farm or greenhouse.

To account for this, the dataset was filtered to include solely houses with the following characteristics:

- Houses with an area between 50 and 350 square metres. An average house in the Netherlands is around 120 square feet, and an average villa is around 200 square feet or larger (de Groot, 2018). The range is flexible to allow for smaller, and larger houses.
- Houses with a height less than 12 metres. The floor-to-ceiling height is around 2.6 metres. Houses with 4 to 5 floors remained in the dataset to allow higher houses.

The dataset still contained a wide variety of house types, such as corner houses, terraced houses, semi-detached houses, bungalows, large villas, older, and newer houses.

[8] Outliers are observations that lie far outside the spread of the data. For example, when observing the height of adults, observations such as 120 centimetres or 310 centimetres are considered outliers.

After filtering, there remained 1,115,404 samples in the dataset. Filtering did not exclude outliers and incorrect samples, therefore it was still important to investigate, and (if necessary) remove them (see Section 4.3.4).

4.3.2 *Data Cleaning: Removing Duplicates*

The second step in the cleaning process involved removing duplicates. One disadvantage of removing duplicates is that it might lead to the loss of valuable information. Conversely, retaining duplicates will result in data leakage, and inflated accuracy scores (Tamilselvi & Gifta, 2011).

In this study, the duplicates represented different parts of the same building (for instance, an extension or shed). Randomly removing these parts would have resulted in a major loss of information. Hence, it was decided to only retain the observation with the largest "height", as the rest of the features in the observation were identical. This strategy removed redundant data, and preserved samples with the most representative information on the houses. After removing the duplicates, 714,353 samples remained in the dataset.

4.3.3 *Data Cleaning: Handling Missing Data*

The dataset missed approximately 0.4% of data in most of the features. Due to time constraints, and the low percentage of missing values, it was decided to remove the missing values. There remained 711,239 samples in the dataset.

Moreover, approximately 51% of the energy labels were missing. In order to utilize the unlabelled data for SSL, the missing labels were filled with the value -1.

4.3.4 *Data Cleaning: Handling Outliers*

The last part of the cleaning process involved dealing with outliers. When it comes to real-world data, handling outliers can be complex for various reasons. In this study, such reasons are the aggregate data from Dutch grid operators, and the previously mentioned errors in the BAG registration.

The aggregated data caused the inclusion of extreme values that did not reflect the true energy, or gas consumption patterns of residents. This problem arises when houses are located near other types of buildings, such as companies, or greenhouses, which consume considerably more energy and gas than the houses themselves. These extreme values were not representative of residential energy consumption, and could negatively influence the performance of the classifier. Figure 6 shows an example where greenhouses caused greatly increased energy consumption values

of the near houses.

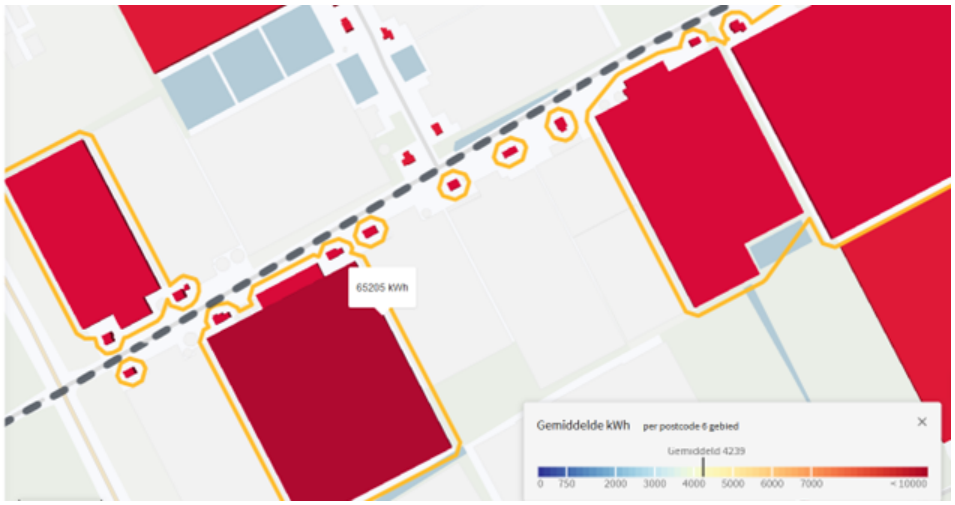

Figure 6: Example of an outlier due to aggregated consumption data. Source: DEGO.vng.nl.

To address outliers, the dataset was filtered using a flexible range to preserve logical varieties of usage patterns between houses. Tables 1 and 2 show the average gas and energy consumption of different types of households. With these values in mind, and still allowing for logical outliers caused by, for example, the year of construction, or area of the building, the dataset was filtered into the following flexible ranges:

- Gas consumption range: [0, 4000]
- Energy consumption range: [0, 6000]

Table 1: Average Yearly Gas and Energy Consumption of Houses with 1 Resident

Type of house	Average gas consumption	Average energy consumption
New small apartment	630 m3	1580 kWh
Old small apartment	800 m3	1550 kWh
Old small house (corner, terraced)	1050 m3	1680 kWh
Old medium-sized house (corner, terraced)	1240 m3	1990 kWh

Source: Milieu Centraal, 2020

Table 2: Average Yearly Gas and Energy Consumption of Houses with 2+ Residents

Type of house	Average gas consumption	Average energy consumption
Old small apartment	1000 m3	2270 kWh
Old small house (corner, terraced)	1220 m3	2790 kWh
Old medium-sized house (corner, terraced)	1380 m3	3200 kWh
New medium-sized house (corner, terraced)	1080 m3	3260 kWh
Old large house (corner, terraced)	1920 m3	3900 kWh
Old large detached house	2400 m3	4530 kWh

Source: Milieu Centraal, 2020

After removing the extreme outliers, 624,530 samples remained in the dataset (refer to Appendix B, page 47, to see the distributions after the removal).

4.3.5 *Preprocessing*

The preprocessing in this study involved feature engineering techniques, such as feature creation, and feature selection. In addition, the data was scaled.

Categorical features

Machine Learning classifiers can easily handle numerical features like height, and energy consumption. However, in the case of categorical features a classifier struggles, especially when there are numerous unique categories. In this study, the categorical feature of interest was the post code feature, which had many unique categories. A common practice in Data Science is to one-hot-encode categorical features, to get them into a numeric form that the classifier prefers. However, when there are many unique categories, one-hot-encoding will lead to increased computational costs due to the high dimensionality of the resulting one-hot-encoded vectors. It was therefore decided to remove the postcode feature from the dataset.

Considering the importance of maintaining spatial relationships in the data, it was decided to create the latitude and longitude degrees at post code level using the available geometric data. Additional created features are explained in the next section.

Feature Creation

It can be useful to experiment with different combinations of features, as a feature by itself might be less meaningful. For instance, when predicting the energy label of a house, it might not be enough to solely rely on features such as height or area. By combining the two, the ***volume*** of each house can be calculated, which can provide more relevant information. This can be achieved by multiplying the area by the height of the house. Feature

creation can help improve the performance of machine learning models by creating more informative features from existing features. In addition to the created ***volume*** feature, the ***kwh_total_p6***, was created by multiplying the energy use by the building count in a post code area. This was done for consistency, as ***gas_total_p6*** was already existent in the dataset.

Feature Selection

The built-in *feature_importances_* attribute from the Random Forest algorithm was used to obtain the feature importance scores of each feature in the dataset (Figure 7). Real-world data often contain redundant and irrelevant features, as many features are included that might not be directly related to the purpose of the research or project (Kotsiantis et al., 2006). Having too many irrelevant and redundant features leads to the so-called curse of dimensionality, and subsequently a decrease in performance and computation speed of the classifier. Hence, removing them is a good practice in machine learning. In this study, features that were redundant, and had a low importance score were removed (*volume_m3_p6*, *grondbeslag_m2_p6*, *kwh_totaal_p6*, *gas_totaal_p6*) from the dataset. The most important features were the non-aggregated features: the construction year, area, volume, and height of the houses.

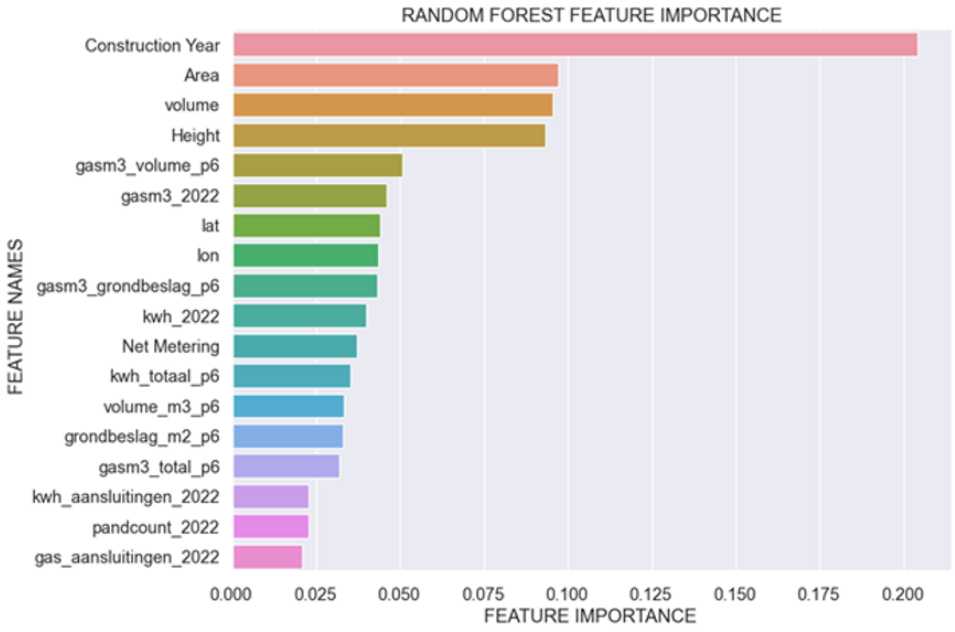

Figure 7: Feature Importances Scores

Label Encoding & Scaling

The final preprocessing steps involved label encoding, and feature scaling. The target labels were encoded using Scikit-Learn's *LabelEncoder* to discard the order between the labels and to provide each individual label a unique numerical label that can be handled by the classifier. The order was discarded to be consistent with the study of Tsoka et al. (2021).

Feature scaling is generally a good practice in Data Science, as it ensures comparable scales across all features, which improves the performance, and computation speed of the classifier (Geron, 2017).

Generally, decision tree-based classifiers do not need feature scaling. However, when a classifier uses a (stochastic) gradient descent algorithm as an optimizer, scaling is crucial to ensure faster convergence to the optimal point (Wan, 2019). XGBoost and TabNet are two examples of classifiers that use gradient descent. Therefore, a robust scaler was applied to the (unlabelled) data.

4.4 *Algorithms*

The algorithms used in this study were Random Forest, XGBoost, and TabNet. This subsection explains the relevant algorithms and their hyperparameters.

4.4.1 *Random Forest*

RFs have been used widely for classification problems and tabular data, and have been favoured for being easy to interpret. As mentioned earlier, the RF algorithm is an ensemble algorithm which stacks multiple independent classifiers. An RF uses the concept of "bagging" to randomly select, with replacement, different samples from the dataset for each individual classifier (Maclin & Opitz, 1997). Each tree in the forest trains independently with its own sampled dataset, and subsequently has its own vote in the final classification of a sample (a house's energy label). When the majority of the trees in the forest, for instance, vote for energy label C, then that is the established class label for the sample (house) in question. A more comprehensive explanation of this algorithm, and its weak classifier, the DT, is provided in Appendix C (page 52).

4.4.2 *XGBoost*

The drawback of a bagging algorithm like RF is that the classifiers train independently of each other, allowing them to make identical errors. This is where XGBoost proves useful.

XGBoost, short for eXtreme Gradient Boosting, is build upon the concept of gradient boosting (Chen & Guestrin, 2016b). While bagging involves stacking independent weak classifiers, boosting combines and adds dependent weak classifiers to pursue better performance. Boosting essentially means successively training new classifiers to correct the errors of the preceding classifier. This process of adding classifiers repeats itself until no further corrections to the previous classifier are possible. Log loss is used as a default loss function.

The designers of XGBoost, Chen and Guestrin (2016b), pointed out the algorithm's success due to its scalability. XGBoost can process a vast amount of data without increasing training time or memory usage. The authors achieved this by including technical innovations to the algorithm and system, such as their so-called *weighted quantile sketch*, and *out-of-core computing* (Chen & Guestrin, 2016b). These innovations prioritize important samples, and enable the processing of large datasets by breaking them into manageable parts. Considering the complexity and size of the dataset used in this study, XGBoost was a suitable fit.

4.4.3 *TabNet*

Recently, Arik and Pfister (2021) proposed the Deep Learning algorithm TabNet, which offers the benefits of explainability and efficiency. In this study, it was chosen to utilize this novel algorithm, and compare its performance with RF, and XGBoost.

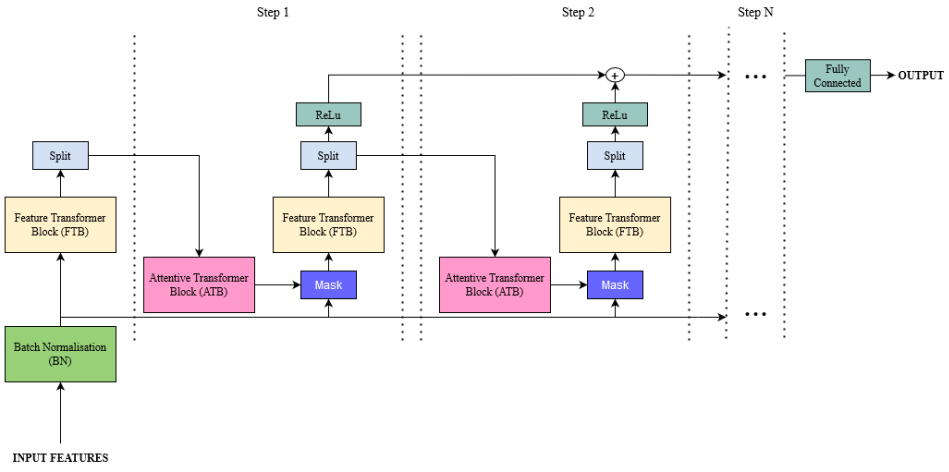

Figure 8: The TabNet Algorithm Simplified - Image made by author, and inspired by the TabNet Paper (Arik & Pfister, 2021).

The TabNet algorithm (Figure 8) is composed of multiple sequential *decision steps*, with each having a vote in the final classification of a sample.

This mimics the concept of ensemble algorithms. The algorithm first applies Batch Normalisation (BN) on the input data, meaning that it standardizes (rescales the data to have a zero mean and one unit variance) the input features to improve efficiency in the training process.

The fundamental components of each decision step are the Attention Transformer Blocks (ATBs), and Feature Transformer Blocks (FTBs). *Attention* is a mechanism in TabNet that assigns extra importance (weight) to the most relevant features for a certain prediction task, which automatically diminishes less important features. This mechanism is supposed to give TabNet its explanatory power, higher accuracy, and efficiency (Arik & Pfister, 2021).

Based on the attention weights, the "Mask" provides information about which features the model selected to make its predictions. The features that were ranked as most important by the ATBs are consequently processed by the FTBs. The FTBs apply several transformations to the features to make them more useful, and to enable the model to learn more complex patterns in the dataset. Examples of the transformations are the BN layers that standardize the features, and the Gated Linear Units (GLU) that apply a so-called activation function, which introduce non-linearity into the model for identifying more complex patterns (Sharma et al., 2022).

TabNet minimizes the error between its predictions and the true labels by the cross-entropy loss function, which is commonly used for multi-class classification problems (Grandini et al., 2020).

4.4.4 *Hyperparameter Tuning*

A Machine Learning model consists of learned parameters (during the training process), and hyperparameters, which are manually set by the model builder. A model, without tuned settings, usually does not result in an optimal performing classifier (Ding et al., 2021). Therefore, model builders experiment with different settings of hyperparameters, which they define through a so-called search space.

There exist numerous tuning methods that try different settings based on the search space. GridSearch tries every possible combination of settings provided in the search space (this is extremely time-consuming). In contrast, RandomSearch randomly selects from the search space, which might not lead to the optimal set of hyperparameters. Therefore, this study employed Optuna, which uses a Bayesian optimisation strategy (Akiba et al., 2019b). This strategy makes informed decisions about which hyperparameters to try next based on the results of the previous trials, resulting in faster, and more effective hyperparameter tuning.

The tuning process is done using a 3-fold stratified cross-validation to mitigate overfitting. The resulting best sets of hyperparameters were evaluated using the test set.

The Optuna optimization process has run for 30 trials for each classifier. This was decided due to time constraints, and after a testing round of 70 trials with the RF Classifier that proved that 30 trials were sufficient to achieve a satisfactory set of hyperparameters (Figure 9).

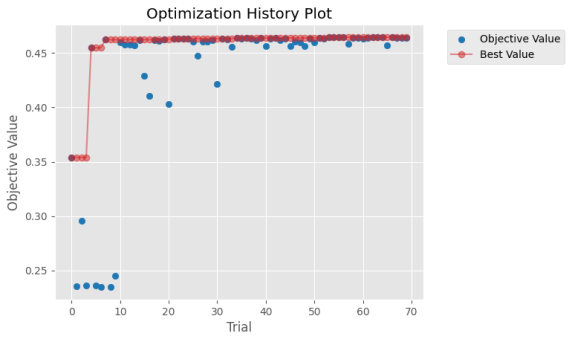

Figure 9: Testing Optimization Process with 70 Trials (RF)

Tuning RF: When tuning the RF, there are a few important hyperparameters to consider. For instance, the ***n_estimators***, and the ***max_depth***, which control the number of trees in the forest, and the depth of the decision tree, respectively. It is important to experiment with reasonable values for each hyperparameter. Extremely high values for ***n_estimators***, and ***max_depth*** would likely result in an overfitting model, as the model becomes too complex. In contrast, setting too low values for these hyperparameters would result in an underfitting model, as it fails to learn the underlying patterns in the data.

The search space for RF (Table 3) was therefore based on logical values and, when available, values recommended by the literature (the same applies to XGBoost and TabNet).

Since the RF had fewer parameters to tune compared to XGBoost and TabNet, it was chosen to experiment with the loss functions[9] here. Gini Impurity, Entropy, and Log Loss are such functions, and suitable for multi-class classification problems. XGBoost, and TabNet used the log loss (cross-entropy) function by default.

Table 3: Hyperparameter Space for Random Forest

Hyperparameter	**Values**
max_depth	2 to 32
n_estimators	100 to 700 (step 50)
max_features	log2, sqrt
criterion	gini, entropy, log_loss
min_samples_leaf	2 to 15 (step 2)
min_samples_split	2 to 15 (step 2)
bootstrap	True, False

Tuning XGBoost: Since XGBoost is likewise a tree-based model, its hyperparameters are similar to those of an RF. The ***n_estimators***, and the ***max_depth*** are set at the same ranges. However, XGBoost has some hyperparameters that are different, such as the ***reg_lambda*** and ***reg_alpha***. These so-called regularization parameters, when set at a higher value, aim to prevent overfitting, and hence the complexity of the model (Zhang et al., 2022). In contrast, if these values are too high, the model might not be able to learn the underlying complex patterns of the data, and hence underfit the data. Table 4 shows the established search space.

Table 4: Hyperparameter Space for XGBoost

Hyperparameter	**Values**
max_depth	2 to 32
n_estimators	100 to 700 (step 50)
eta	0.01 to 0.1 (log uniform)
subsample	0.5 to 1
colsample_bytree	0.5 to 1
reg_alpha	0.001 to 10 (log uniform)
reg_lambda	0.001 to 10 (log uniform)
gamma	0 to 1 (step 0.01)
early_stopping_rounds	10

[9] A loss function is used to minimize the difference between the true labels, and the labels predicted by the classifier.

Early stopping (XGBoost), and patience (TabNet) were applied to prevent overfitting, and to stop the training when the scores stopped improving.

Tuning TabNet: The TabNet algorithm includes a variety of hyperparameters, such as, ***N_steps*** which controls the number of decision steps, and ***n_a*** which controls the size of the attention mechanism.

Arik and Pfister (2021) provided recommendations on how to set the search space for the hyperparameters. For instance, they recommended a range from 3 to 10 for the hyperparameter ***N_steps***, as higher values might cause overfitting. In addition, they supported setting ***n_a*** equal to ***n_d***, again keeping in mind that higher values may cause overfitting, and hence poorer generalization of the model. Finally, they recommended a large batch size (1% of the training set size), and a high learning rate (0.02) at the start, which should gradually decrease to the point of convergence [10]. Table 5 shows the established search space.

Table 5: Hyperparameter Space for TabNet

Hyperparameter	**Values**
mask_type	entmax, sparsemax
n_da	8 to 32 (step 8)
n_steps	3 to 10 (step 1)
gamma	0.01 to 0.2 (step 0.01)
n_shared	1, 2, 3
lambda_sparse	1e-6 to 1e-3 (log scale)
lr (learning rate)	starting from 0.02
patienceScheduler (lr)	3, 4, 5, 6
patience (early stopping)	4, 5, 6, 7, 8
max_epochs	10 to 50

[10] Convergence means that the model is at its optimum performance at a certain point and will no longer improve after that certain point.

4.5 *Experimental Set-up*

This subsection elaborates on the experiments conducted in order for the research questions to be answered.

Experiment 1: Tuning with imbalanced data
In the first experiment, the classifiers RF, XGBoost and TabNet were tuned with the unbalanced data. This experiment was conducted to investigate the impact of the class imbalance on the classifiers, and whether the tuned parameters improve or worsen performances.

Experiment 2: Tuning with resampled data
The second experiment evaluated the effectiveness of SMOTE-Tomek in improving the performance of classifiers, and overcoming the class imbalance by retuning the models on resampled data. SMOTE-Tomek is a combination of SMOTE, and Tomek Links. As mentioned in the literature review (Section 3.2), SMOTE generates synthetic data to create more samples for the minority classes. The disadvantage of SMOTE on its own, is the possibility that it can create noisy, and overlapping samples. The Tomek part aimed at removing overlapping samples between classes. Since the dataset used in this work contained aggregated data and a wide range of energy classes, overlapping observations were expected. It was therefore decided to apply SMOTE-Tomek to the training set, and subsequently evaluate its ability to resolve the class imbalance, and reduce overlapping samples.

Experiment 3: Semi-Supervised Learning
In the third and final experiment, semi-supervised learning was used to utilize unlabelled data, and potentially improve the performance of the best-performing model resulting from experiments 1, and 2. The general idea behind SSL is to train a classifier based on both labelled and unlabelled data. Several types of Semi-Supervised Learning methods exist, such as self-training, and graph-based methods like label propagation (Prakash & Nithya, 2014). This study used the ***LabelPropagation*** algorithm, as it is a graph-based method that uses a distance metric to propagate labels. This approach was preferred over solely using a pre-trained supervised classifier for label estimation, as the supervised classifier might not be reliable enough to correctly predict the energy labels of houses. Therefore, self-training was not considered appropriate for this experiment.

As shown in Figure 10, the label propagation algorithm creates a connected graph with nodes and edges. The nodes represent the labelled and unlabelled samples in the dataset, and the edges visualize the similarities

between these samples based on the Euclidean distance metric (Chapelle et al., 2006). Via the connected graph, the labels are propagated to the unlabelled samples.

The mixed labelled and unlabelled datasets, and the subsequently derived propagated labels, are fed to the best-performing classifier to evaluate whether the additional data can further improve the performance of the classifier.

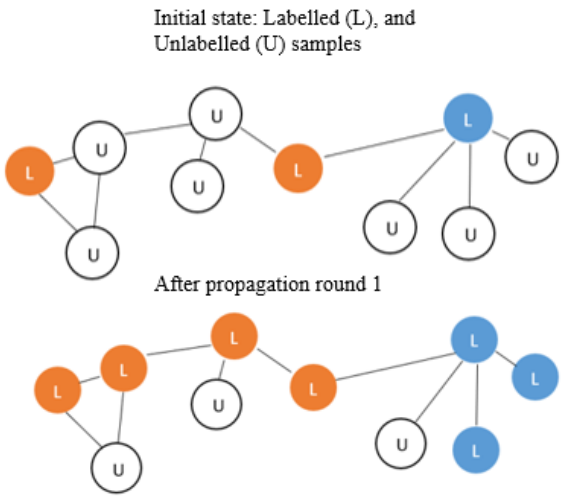

Figure 10: Visualizing the Label Propagation Algorithm

4.6 *Evaluation Methods*

This thesis dealt with a multi-class classification problem, with severe class imbalances. During the evaluation of the classifiers, it must be clear how the classifiers perform per class, in order to draw proper conclusions about the overall performances.

In this thesis, it was not sufficient to rely solely on *Accuracy* as an evaluation method, since a classifier subjected to unbalanced data is likely to frequently predict the majority class as the outcome, leading to an incorrectly high accuracy score. Therefore, this thesis used the *Balanced Accuracy Score*, the *macro precision*, *macro recall*, *macro F1*, and *Cohen's Kappa* scores as evaluation metrics. The scores are based on the well-known *Confusion Matrix*.

Confusion Matrix
A confusion matrix compares the true classes with the predicted classes. There are four situations which can appear, namely:

- **True Positives (TP)**: a correctly predicted the positive class
- **False Positives (FP)**: an incorrectly predicted the positive class
- **True Negatives (TN)**: a correctly predicted the negative class
- **False Negatives (FN)**: an incorrectly predicted the negative class

These categories are based on a binary classification problem, however the confusion matrix, and its associating evaluation metrics can be extended to a multi-class problem. The Precision, and Recall metrics can be calculated per class, and then macro-averaged to obtain the score for the entire classifier.

Macro precision, recall, F1, and Balanced Accuracy
Precision and recall are important metrics, as they give an indication how the classifier is performing per class. Macro refers to the averaging method that is used for calculating the scores. Macro averaging weights each class equally, regardless of its size. This ensures that the actual performance of the classifier is considered across all classes. The formulas are shown in Equation's 1, and 2, where n indicates the number of classes, and i indicates the particular class being calculated. Afterwards, all the classes calculated are being averaged[11]. All scores rank from 0 to 1.

[11] Note that the confusion matrices in this study are presented normalised for readability. The calculations should be performed without the normalisation.

$$\text{Macro Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i} \tag{1}$$

$$\text{Macro Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i} \tag{2}$$

When evaluating the classifier's performance on predicting the class F, the *precision* score reveals: **the percentage of correctly predicted F labels out of all predicted F labels** (Figure 11).

The recall score reveals **the percentage of correctly predicted F labels out of the actual F labels** (Figure 11).

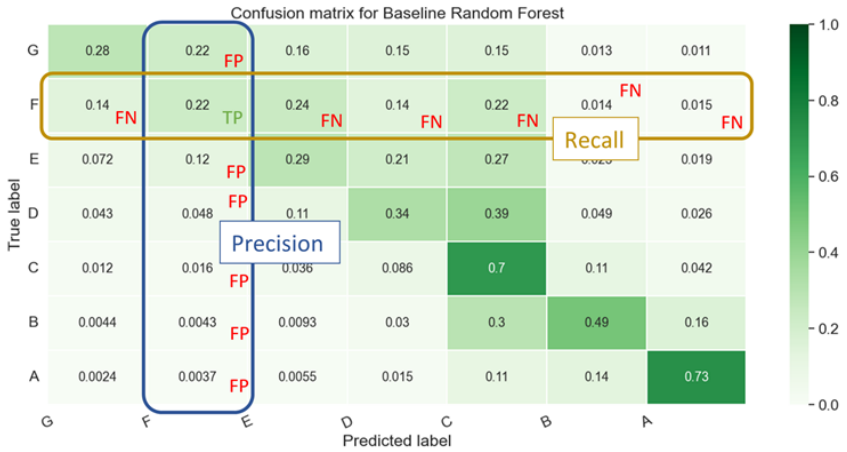

Figure 11: Visualizing Relevant Values for Precision and Recall for Class F

In order to optimize the performances of the classifiers, the *macro F1-Score* was chosen as the optimization value during the tuning process in Optuna. It is a suitable evaluation metric, as it takes into account both precision and recall, which were both important scores for predicting energy labels. The Macro F1 is calculated as shown in Equation 3. The overall performances, and the performances per class of each classifier will be evaluated, and compared against the baseline classifier, and each other.

$$\text{Macro F1} = 2 \cdot \frac{\text{macro precision} \cdot \text{macro recall}}{\text{macro precision} + \text{macro recall}} \tag{3}$$

The balanced accuracy (Equation 4), which ranks from 0 to 1, takes into account the class imbalance by averaging the recall scores for each class and subsequently applies the arithmetic mean ($\frac{1}{n}\sum_{i=1}^{n}$) of these scores to compute the balanced accuracy (Grandini et al., 2020; Pedregosa et al., 2011).

$$\text{Balanced Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i} \tag{4}$$

Cohen's Kappa

The Cohen's Kappa score is used to evaluate the agreement between the ground truth labels and the labels predicted by the classifier. When the ground truth equals the predicted label, there is agreement, if not, there is disagreement. Cohen's Kappa takes agreement by chance into consideration. The score ranges from -1 to 1, where -1 means the worst performing classifier, 0 means the classifier is simply guessing, and 1 indicates a perfect classifier. The formula is shown in Equation 5, where p_0 indicates the observed agreement, and p_e indicates the expected agreement.

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{5}$$

5 RESULTS

This section reports the results of the experiments conducted with Random Forest, XGBoost, and TabNet.

5.1 *Results Experiment 1: Tuning with imbalanced data*

As presented in Tables 7, 8, and 9, each baseline model showed relatively low precision, recall, and F1-scores. The F1-scores for RF, XGBoost, and TabNet were 0.442, 0.447, and 0.376 respectively. The Confusion Matrices (Figures 27, 31, 35, in Appendix E, F, and G) depict that the baselines were, as expected, sensitive to the majority class C. Especially TabNet predicted class C for a considerable amount of houses. The performances per class (Tables 12, 15, 18, in Appendices E, F, and G respectively) reveal that all the baselines performed particularly poor (F1-scores between 0.11, and 0.36) in classifying the classes D to G. The precision, and recall scores for classes A, and C were considerably higher, which is comprehensible due to the class imbalance.

The Cohen's Kappa scores for the baselines were around 0.44 for RF, and XGBoost, and 0.36 for TabNet. These scores indicate that there was a moderate agreement. The poor scores for each classifier might have been caused by the class imbalance, or the fact that the classifiers were untuned.

Table 6: Optimal Hyperparameter Set per Classifier after 30 Trials

Model	**Optimal Hyperparameter Set (30 trials)**
RF	max_depth = 28, n_estimators = 200, max_features = sqrt, criterion = log_loss, min_samples_leaf = 5, min_samples_split = 3, bootstrap = False
RF + SMOTE-Tomek	max_depth = 20, n_estimators = 500, max_features = sqrt, criterion = entropy, min_samples_leaf = 12, min_samples_split = 6, bootstrap = False
XGBoost	max_dept = 24, n_estimators = 600, eta = 0.0358, subsample = 0.9609, colsample_bytree = 0.9287, reg_alpha = 0.0101, reg_lambda = 9.3487, gamma = 0.820
XGBoost + SMOTE-Tomek	max_dept = 11, n_estimators = 450, eta = 0.0191, subsample = 0.8083, colsample_bytree = 0.8972, reg_alpha = 0.0031, reg_lambda = 1.2791, gamma = 0.97
TabNet	mask_type = entmax, n_a = 24, n_d = 24, n_steps = 10, gamma = 0.069, n_shared = 2, lambda_sparse = 3.4645, patienceScheduler = 6, patience = 8, max_epochs = 49
TabNet + SMOTE-Tomek	mask_type = entmax, n_a = 24, n_d = 24, n_steps = 7, gamma = 0.12, n_shared = 2, lambda_sparse = 5.7601, patienceScheduler = 5, patience = 8, max_epochs = 22

Therefore, it was decided to first tune all the classifiers with the imbalanced data, and see how they perform. The best sets of hyperparameters for each classifier after 30 trials are presented in Table 6. The hyperparameter settings demonstrate a balance between capturing complex patterns in the data, and optimizing performance, without increasing the risk of overfitting. The Optuna optimization plots can be found in Appendices E, F, and G.

The classifiers that were tuned on the imbalanced data, showed minor improvements in F1-scores (Tables 7, 8, and 9). RF, and XGBoost increased by approximately 2%, while TabNet increased by 3%. Especially RF increased performances in classifying the minority classes with precision, and recall scores increased by approximately 3% on average (Table 13, Appendix E).

Tables 13, 16, and 19 (Appendices E, F, and G) reveal that the F1-scores for the minority classes now ranged from 0.15 to 0.37, with class F having the lowest F1-score among all tuned classifiers. Class A had the highest F1-score for each tuned classifier (ranging from 0.73 to 0.78).

Table 7: Random Forest with and without SMOTE-Tomek

Model	Precision	Recall	F1-Score	Balanced Acc	Cohen's Kappa
Baseline RF	0.451	0.436	0.442	0.436	0.440
Tuned RF	**0.484**	0.454	0.464	0.454	**0.467**
Tuned RF + SMOTE-Tomek	0.469	**0.507**	**0.478**	**0.507**	0.464

Table 8: XGBoost with and without SMOTE-Tomek

Model	Precision	Recall	F1-Score	Balanced Acc	Cohen's Kappa
Baseline XGB	**0.483**	0.440	0.447	0.440	0.442
Tuned XGB	0.478	0.453	0.462	0.453	**0.475**
Tuned XGB + SMOTE-Tomek	0.474	**0.506**	**0.483**	**0.506**	0.466
XGB + Semi-supervised	0.464	0.498	0.472	0.498	0.452

Table 9: TabNet with and without SMOTE-Tomek

Model	Precision	Recall	F1-Score	Balanced Acc	Cohen's Kappa
Baseline TabNet	0.443	0.375	0.376	0.375	0.364
Tuned TabNet	**0.448**	0.405	**0.409**	0.405	**0.404**
Tuned TabNet + SMOTE-Tomek	0.407	**0.448**	0.405	**0.448**	0.351

The Cohen's Kappa scores for RF, XGBoost, and TabNet increased to 0.47, 0.48, and 0.40, respectively. These scores indicate that the agreement in predicting energy labels was stronger than would have been expected based on chance alone. However, there remained substantial disagreements, and plenty of room for improvements.

The persistently low scores were likely due to the still present class imbalance, indicating that the tuned classifiers remained sensitive to the majority class C. The next section examines whether SMOTE-Tomek counteracts the class imbalance, and improves the ability of the classifiers to correctly predict the minority classes.

5.2 *Results Experiment 2: Tuning with balanced data (SMOTE-Tomek)*

Tables 7, 8, and 9 show that the RF, and XGBoost classifiers with SMOTE-Tomek improved marginally in overall F1-scores (+1.4% and +2.1% respectively) compared to the classifiers without SMOTE-Tomek. The F1 score for TabNet dropped a very slight 0.4%. Notably, the precision scores declined for each classifier, indicating that there were more false positive predictions.

On average, the recall scores for classes E-G demonstrated substantial improvements across RF, XGBoost, and TabNet, with increases of 13%, 12%, and 15%, respectively (Tables 14, 17, 20, Appendices E, F, and G).

In contrast, The recall scores for the former majority classes C, and A decreased by an average of 11%, 10%, and 25% for RF, XGBoost, and TabNet, respectively. These performance decreases may be due to the Tomek part of SMOTE-Tomek, which might have removed too many important samples on which the classifiers relied.

The Confusion Matrices in Figures 29, 33, 37 (Appendices E, F, and G) demonstrate that most of the incorrect predictions shifted from class C (without SMOTE-Tomek data), to classes E, F, and G (with SMOTE-Tomek data). This shift can be attributed to the resampling of the training data with SMOTE-Tomek, with classes E, F and G becoming slightly more dominant, as shown in figure 12.

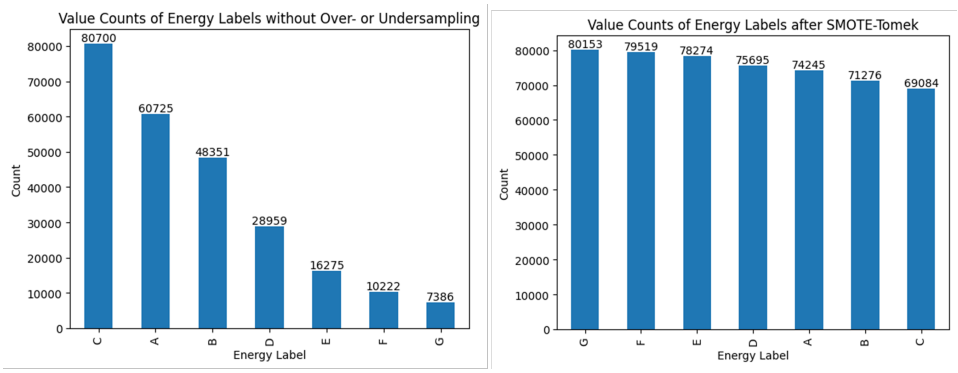

Figure 12: Distribution of the Class Labels before, and after Merging

The Cohen's Kappa scores decreased slightly to 0.46, 0.47, and 0.35, for RF, XGBoost, and TabNet, respectively. This suggests that the SMOTE-Tomek classifiers were guessing the E, F, and G classes more often, leading to decreased agreement, and a higher number of false predictions, which is in line with the declined precision scores.

Confusion among classifiers between closely related classes may have contributed to the drop in the precision, and Kappa scores. This could be because various houses with different energy labels share certain data

characteristics, making it challenging for classifiers to distinguish between them. This may have been intensified through the data generated with SMOTE. The confusion matrices (Figures 29, 33, 37, Appendices E, F, and G) reveal frequent misclassifications, including instances where class G was predicted as F or E, class F was predicted as E or G, and class E was predicted as F or D.

Overall, the classifiers showed marginal improvements with SMOTE-Tomek, as evidenced by the increased F1-scores, and Balanced Accuracy scores in Tables 7, 8, and 9 (except for TabNet's F1-score). The performances for the minority classes improved for each classifier, with F1-scores now ranging from 0.26 to 0.41. Class F remained the class with lowest F1-score among all classifiers. Class A remained the best predicted classes for each classifier (F1-scores ranging from 0.71 to 0.77).

RF, and XGBoost outperformed TabNet with similar performances. However, it was decided to declare the tuned XGBoost classifier with SMOTE-Tomek as best-performing, since its highest Macro F1-score. The following section examines whether leveraging unlabelled data could further improve the XGBoost classifier.

5.3 *Results Semi-Supervised Learning*

The best-performing XGBoost classifier was re-trained with 308,757 additional unlabelled samples, resulting in a slight decrease (-1.1%) in F1-score (Table 8). Tables 10, and 17 (Appendix F) demonstrate that the recall and precision scores per class increased or decreased with approximately 1% or 2% as compared to the scores of the XGBoost classifier without additional training data. These increases or decreases, however, were not considered substantial for each class.

The overall decreased performance could be attributed to increased noise introduced by the label propagation algorithm. This noise likely has arisen from the propagation of unreliable labels, which may have been a result of the confusion between closely related classes during the propagation process. As a result, the label propagation algorithm may have propagated labels incorrectly, leading to increased complexity during training, and subsequently a slight decline in XGBoost's performance. Figure 13 proves the continued confusion between closely related energy labels.

Table 10: Performance per Class Best-performing XGBoost + SMOTE-Tomek + SSL

Class	Precision	Recall	F1-Score	Support
G	0.28	0.48	0.35	1846
F	0.24	0.35	0.29	2555
E	0.30	0.41	0.35	4069
D	0.37	0.41	0.39	7240
C	0.65	0.60	0.63	20175
B	0.56	0.54	0.55	12088
A	0.85	0.68	0.76	15182

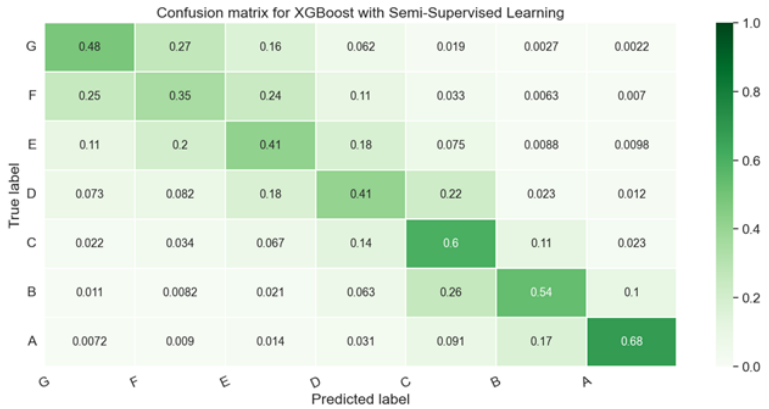

Figure 13: Confusion Matrix for Best-performing XGBoost + SMOTE-Tomek + SSL

5.4 *Visualizing Predictions with QGIS*

A potential reason for the presumably overlapping characteristics in the data, and the poor performances of the classifiers, is visualized with QGIS in Figure 14. From the figure, it appears that, for instance, the XGBoost classifier struggled with the aggregated data from the grid operators, as it seems to predict class E for an entire street or postcode area. It was unable to correctly predict the deviating houses with energy label D, and B.

Figure 14: Test set results - True Labels and Predicted Labels by the best XGBoost model With SMOTE-Tomek

Another possible reason is the absence of data regarding individual house-level attributes, such as data on the building envelope. Such information is commonly considered in other studies as mentioned in 3.1.2, and unfortunately not available in this thesis. As a result, the classifiers might struggle to differentiate between energy labels due to overlapping characteristics, and the lack of this additional data.

Additional figures with true, and predicted labels can be found in Appendix H (page 64).

5.5 *Feature Importances TabNet*

TabNet's relatively weaker performance compared to the other classifiers may be attributed to variations in its emphasis on different features. Figures 15, and 16 show that the RF laid substantially more emphasis on the construction year than TabNet did. TabNet placed equal emphasis on the rest of the features, while RF, for instance, considered gas and energy connections (Dutch: aansluitingen) to be less important. These different views on the features may have caused TabNet's performance to be poorer.

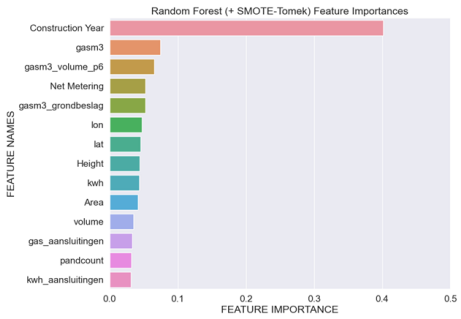

Figure 15: Feature Importance Random Forest (+SMOTE-Tomek)

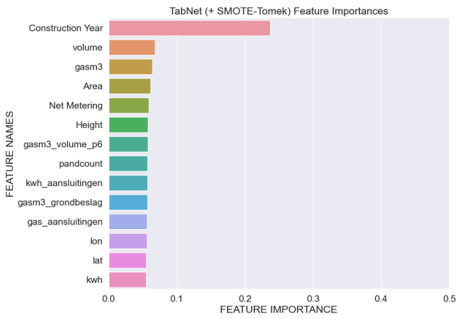

Figure 16: Feature Importance TabNet (+SMOTE-Tomek)

6 DISCUSSION

The goal of this thesis was to evaluate to what extent Random Forest, XGBoost, and TabNet were able to predict the energy labels of houses. There were several challenges in this study, such as the imbalance between classes, unlabelled samples, aggregated data from grid operators, and the absence of data on, for instance, the building envelope.

6.1 *Results Discussion*

The results of this study indicated that the baseline classifiers were considerably poor at predicting energy labels of houses. The first experiment examined whether that was due to the classifiers' untuned parameters. The performance of the subsequent tuned classifiers improved slightly, however, the models remained poor at predicting the minority classes. This was likely due to the class imbalance that still existed.

Therefore, in experiment 2, the classifiers were retuned with SMOTE-Tomek applied to the training data to smooth the class imbalance. This resulted in minor improvements in the overall performances of the classifiers (except for an extremely minor decrease for TabNet), which is consistent with the result of the studies from Cai et al. (2019) and Le et al. (2018). The minority classes, for each classifier, were now predicted considerably better. Contrastingly, the performances in predicting the former majority classes (A, and C) dropped. This was likely caused by the shift in the class distribution, making classes A, B and C the slight [12] minority classes. Additionally, the reduction could be attributed to the potential loss of information due to Tomek's undersampling, as stated in the study of (Yang & Li, 2022).

[12] Note the word *slight* here, as the class imbalance was no longer severe.

Experiment 3 investigated whether additional unlabelled data could further improve the performance of the best-performing XGBoost classifier. Unfortunately, the additional data slightly decreased the performance. The label propagation algorithm relies on similarities, as stated by (Ligthart et al., 2021), and (Van Engelen & Hoos, 2019). Due to potentially overlapping characteristics between classes due to aggregated features in the dataset, the algorithm might have been confused during the label propagation process, resulting in a decrease in the performance of XGBoost.

What contradicted with the literature and went against expectations were the relatively poor performances of the TabNet classifiers (Arik & Pfister, 2021). TabNet might have benefitted from a larger dataset. Although, remarkably, Tsoka et al. (2021) achieved more success with an artificial neural network, even when working with a smaller dataset. Therefore, the poor performances might also have been due to the varying feature importance scores, or the aggregated data from grid operators.

The overall poor performances of the classifiers were expected to stem from the (lack of) input data. Cai et al. (2019) and Tsoka et al. (2021), for instance, had access to non-aggregated data, and data on the building envelope, and emphasised the importance of such data in providing more reliable predictions.

6.2 *Limitations*

This study had several limitations that should be acknowledged. Firstly, the unavailability of gas and energy data information on individual house-level. The aggregated data from the grid operators might have restricted the classifiers' ability to obtain detailed insights.

Secondly, the absence of information on the building envelope, and indoor conditions, such as the presence of heat pumps or solar panels in a house, presumably limited the potential improvement of the classifiers' performances. Additional data could have provided more detailed insights at the individual house level, which was likely lacking in this study.

Lastly, given time constraints, it was decided to use the Brabant dataset instead of the substantially larger Netherlands (NL) dataset. This decision was made to reduce computation time, as working with the larger dataset would have greatly increased the processing time for each classifier and its tuning process. Appendix I (page 65) informs that the NL dataset did not substantially increase the performances of the tuned RF, and XGBoost.

6.3 *Relevance*

As mentioned earlier, limited studies have been done on predicting energy labels. This study used previously unexplored data to investigate the feasibility of energy label prediction. The findings demonstrated that XGBoost slightly outperformed RF, and TabNet. Additionally, SMOTE-Tomek proved effective in improving the predictive performance in classifying the minority classes. Unfortunately, the poor performance of all three classifiers hampers the reliable use of predicted energy labels for effective transition planning and policy interventions for now. Nevertheless, this downside underlines the importance of useful input data for accurate predictions of energy labels, which is in line with observations in previous studies (Section 3.1.2) (Olu-Ajayi et al., 2022; Tsoka et al., 2021). This emphasizes the importance for the VNG to consider including additional non-aggregated data, and data on the building envelope in future research to potentially improve prediction performances by including more comprehensive data on the houses, if allowed by privacy laws.

6.4 *Future Work*

There are a few options for follow-up research. Firstly, it would be valuable to explore whether the use of non-aggregated data enhances the performance of the classifiers. Secondly, it could be opted to add supplemental input features that could provide more insight into the characteristics of the houses regarding, for instance, the walls, and windows. Similar to the study of Tsoka et al. (2021), explainable AI methods could be used to investigate which features would have the most impact on the classifiers.

Third, the application of ordinal classification could be considered to account for the order between energy labels. Given the potential confusion between energy labels in this study, the use of ordinal classification techniques could further enhance prediction accuracy, as the penalty of a misclassified sample gets higher as the distance between the predicted, and true label expands (Kim et al., 2016). This could enhance the predictive performance of the classifiers.

7 CONCLUSION

This section answers the research question of this study.

7.1 *Sub-RQ 1: How do Machine Learning algorithms (RF & XGBoost), and Deep Learning algorithm TabNet perform in predicting the energy labels of houses with imbalanced data?*

In terms of overall macro F1 and balanced accuracy scores, the tuned RFs and XGBoost classifiers demonstrated similar performance. RF achieved an F1 score and balanced accuracy of 0.464 and 0.454, while XGBoost achieved 0.462 and 0.453. In contrast, the TabNet algorithm performed the worst, with an F1 score of 0.409 and balanced accuracy of 0.405. This could be attributable to TabNet's extreme sensitivity to majority class C, even after tuning. Overall, the classifiers had difficulties in accurately predicting the energy labels of houses, especially for the minority classes.

7.2 *Sub-RQ 2: Does SMOTE-Tomek overcome the class imbalance, and improve the performances of the classifiers?*

The application of the SMOTE-Tomek resampling technique proved beneficial for the classifiers. In particular, the recall scores for the minority classes showed substantial improvements, with increases of more than 10% for each classifier. In contrast, the performances for the formerly majority classes deteriorated among all classifiers. Together, these findings indicated that SMOTE-Tomek effectively addressed the class imbalance and resulted in marginally better performance for the minority classes. Despite this progress, the classifiers remained poor at accurately predicting energy labels of houses.

7.3 *Sub-RQ 3: Can Semi-supervised Learning improve the prediction performance of the best-performing model via exploiting unlabelled samples in the dataset?*

Unfortunately, utilizing unlabelled data through semi-supervised learning did not result in an improvement in the performance of the best-performing XGBoost classifier, which was tuned with SMOTE-Tomek. The nature of the data presumably troubled the label propagation algorithm in correctly differentiating between classes, and propagating the labels.

7.4 *Overarching RQ: To what extent can Machine Learning algorithms predict the energy labels of houses in the province Noord-Brabant of the Netherlands?*

Based on the findings in this study, the algorithms had difficulties in accurately predicting the energy labels of houses in Noord-Brabant. RF, and XGBoost demonstrated similar performances. However, TabNet performed slightly worse compared to RF, and XGBoost. The use of SMOTE-Tomek did improve the abilities of each classifier to predict the minority classes slightly more accurately, however the overall performance remained relatively poor. Moreover, utilizing unlabelled data through semi-supervised learning did not result in improvement in performance for the best-performing XGBoost classifier. To conclude, although the Machine Learning algorithms overall showed considerable potential in predicting energy labels, there is still room for improvements which can be achieved by considering the recommendations for further research.

REFERENCES

Abney, S. (2004). Understanding the yarowsky algorithm. *Computational Linguistics*, *30*(3), 365–395.

Aghajanzadeh, Sercan. (2019). PyTorch TabNet.

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019a). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019b). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

Arik, S. Ö., & Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(8), 6679–6687.

Barreto, C. A., Gorgônio, A. C., Canuto, A. M., & Xavier-Júnior, J. C. (2020). A distance-weighted selection of unlabelled instances for self-training and co-training semi-supervised methods. *Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part II*, 352–366.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Cai, H., Shen, S., Lin, Q., Li, X., & Xiao, H. (2019). Predicting the energy consumption of residential buildings for regional electricity supply-side and demand-side management. *IEEE Access*, *7*, 30386–30397.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. MIT Press.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chen, T., & Guestrin, C. (2016a). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., & Guestrin, C. (2016b). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

de Groot, N. (2018). Een kast van een huis is niet meer van deze tijd: We wonen steeds kleiner. *Algemeen Dagblad*. https://www.ad.nl/economie/een-kast-van-een-huis-is-niet-meer-van-deze-tijd-we-wonen-steeds-kleiner%5C~a20f8b5d/

development team, T. P. (2020). *Pandas-dev/pandas: Pandas* (Version latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Ding, Y., Fan, L., & Liu, X. (2021). Analysis of feature matrix in machine learning algorithms to predict energy consumption of public buildings. *Energy and Buildings*, *249*, 111208.

Dutch Government. (2019). Klimaatakkoord. Retrieved February 3, 2023, from https://www.klimaatakkoord.nl/documenten/publicaties/2019/06/28/klimaatakkoord

Fathi, S., Srinivasan, R., Fenner, A., & Fathi, S. (2020). Machine learning applications in urban building energy performance forecasting: A systematic review. *Renewable and Sustainable Energy Reviews*, *133*, 110287.

Geron, A. (2017). *Hands-on machine learning with scikit-learn and tensorflow : Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Gillies, S., et al. (2007). Shapely: Manipulation and analysis of geometric objects. *toblerity.org*. https://github.com/Toblerity/Shapely

González-Briones, A., Hernández, G., Pinto, T., Vale, Z., & Corchado, J. M. (2019). A review of the main machine learning methods for predicting residential energy consumption. *2019 16th International Conference on the European Energy Market (EEM)*, 1–6.

Goyal, M., Pandey, M., & Thakur, R. (2020). Exploratory analysis of machine learning techniques to predict energy efficiency in buildings. *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, 1033–1037.

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: An overview. *arXiv preprint arXiv:2008.05756*.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jordahl, K., den Bossche, J. V., Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., … Leblanc, F. (2020). *Geopandas/geopandas: Vo.8.1* (Version vo.8.1). Zenodo. https://doi.org/10.5281/zenodo.3946761

Kim, S., Kim, H., & Namkoong, Y. (2016). Ordinal classification of imbalanced data with application in emergency and disaster information services. *IEEE Intelligent Systems*, *31*(5), 50–56.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International journal of computer science*, *1*(2), 111–117.

Kovács, G. (2019). Smote-variants: A python implementation of 85 minority oversampling techniques. *Neurocomputing*, *366*, 352–354.

Le, T., Lee, M. Y., Park, J. R., & Baik, S. W. (2018). Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset. *Symmetry*, *10*(4), 79.

Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. *Applied Soft Computing*, *101*, 107023.

Lindsey, R., & Dahlman, L. (2023). Climate change: Global temperature. Retrieved February 21, 2023, from https://www.climate.gov/news-features/understanding-climate/climate-change-global-temperature

Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. *AAAI/IAAI*, *1997*, 546–551.

Naber, D., & Miłkowski, M. (2005, August 15). *Languagetool* (Version 6.2.5). https://languagetool.org

Olu-Ajayi, R., Alaka, H., Sulaimon, I., Sunmola, F., & Ajayi, S. (2022). Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. *Journal of Building Engineering*, *45*, 103406.

OpenAI. (2021). ChatGPT: Language model for conversational agents.

Papadopoulos, S., Azar, E., Woon, W.-L., & Kontokosta, C. E. (2018). Evaluation of tree-based ensemble learning algorithms for building energy performance estimation. *Journal of Building Performance Simulation*, *11*(3), 322–332.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2013). *Scikit-learn* (Version 1.2.1). https://scikit-learn.org

Pérez-Lombard, L., Ortiz, J., & Pout, C. (2008). A review on buildings energy consumption information. *Energy and buildings*, *40*(3), 394–398.

Pollet, T. V., Stulp, G., Henzi, S. P., & Barrett, L. (2015). Taking the aggravation out of data aggregation: A conceptual guide to dealing with statistical issues related to the pooling of individual-level observational data. *American journal of primatology*, *77*(7), 727–740.

Prakash, V. J., & Nithya, D. L. (2014). A survey on semi-supervised learning techniques. *arXiv preprint arXiv:1402.4645*.

QGIS Development Team. (2009). *Qgis geographic information system*. Open Source Geospatial Foundation. http://qgis.org

Rijksoverheid. (2014). Waarom een verplicht energielabel? Retrieved February 3, 2023, from https://www.rijksoverheid.nl/onderwerpen/energielabel-woningen-en-gebouwen/waarom-een-verplicht-energielabel

Rijksoverheid. (2016). Energielabels van woningen.

Sasada, T., Liu, Z., Baba, T., Hatano, K., & Kimura, Y. (2020). A resampling method for imbalanced datasets considering noise and overlap. *Procedia Computer Science*, *176*, 420–429.

Seyedzadeh, S., Rahimian, F. P., Rastogi, P., & Glesk, I. (2019). Tuning machine learning models for prediction of building energy loads. *Sustainable Cities and Society*, *47*, 101484.

Seyrfar, A., Ataei, H., Movahedi, A., & Derrible, S. (2021). Data-driven approach for evaluating the energy efficiency in multifamily residential buildings. *Practice Periodical on Structural Design and Construction*, *26*(2), 04020074.

Sharma, J., Maheshwari, R., Khan, S., & Ali, A. (2022). Evaluating performance of different machine learning algorithms for the acute emg hand gesture datasets. *Journal of Electronics*, *4*(3), 192–201.

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90.

Tamilselvi, J. J., & Gifta, C. B. (2011). Handling duplicate data in data warehouse for data mining. *International Journal of Computer Applications*, *15*(4), 7–15.

Thesaurus. (2023). In *Thesaurus.com*. https://www.thesaurus.com/

Tsoka, T., Ye, X., Chen, Y., Gong, D., & Xia, X. (2021). Building energy performance certificate labelling classification based on explainable artificial intelligence. *Neural Computing for Advanced Applications: Second International Conference, NCAA 2021, Guangzhou, China, August 27-30, 2021, Proceedings 2*, 181–196.

Tsoka, T., Ye, X., Chen, Y., Gong, D., & Xia, X. (2022). Explainable artificial intelligence for building energy performance certificate labelling classification. *Journal of Cleaner Production*, *355*, 131626.

United Nations. (2019). Paris agreement. Retrieved February 3, 2023, from https://unfccc.int/sites/default/files/english_paris_agreement.pdf

Van Engelen, J. E., & Hoos, H. H. (2019). A survey on semi-supervised learning. *Machine learning*, *109*(2), 373–440.

VNG. (2020). Transitievisie warmte. Retrieved February 5, 2023, from https://vng.nl/artikelen/transitievisie-warmte

Walker, S., Khan, W., Katic, K., Maassen, W., & Zeiler, W. (2020). Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings. *Energy and Buildings*, *209*, 109705.

Wan, X. (2019). Influence of feature scaling on convergence of gradient iterative algorithm. *Journal of physics: Conference series*, *1213*(3), 032021.

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Yang, H., & Li, M. (2022). Software defect prediction based on smote-tomek and xgboost. *Bio-Inspired Computing: Theories and Applications: 16th International Conference, BIC-TA 2021, Taiyuan, China, December 17–19, 2021, Revised Selected Papers, Part II*, 12–31.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *33rd annual meeting of the association for computational linguistics*, 189–196.

Zhang, P., Jia, Y., & Shang, Y. (2022). Research and application of xgboost in imbalanced data. *International Journal of Distributed Sensor Networks*, *18*(6), 15501329221106935.

Zhu, X., & Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation.

APPENDIX A

Table 11: Description of features

Feature name	Description
Area	The area in square metres of the building
Construction Year	The year in which the house was built
Energy Label	Energy label of the house
function_score	Ranging from 0 to 100%. 0 is business only, 100 is residential only
gas_m^3_2022	Average yearly gas use per cubic meter per house in 2022
gas_connections	Number of gas connections in postcode area
gas_m^3_grondbeslag	Total gas consumption of a group of houses offset by area m2
gas m^3 total	Total gas consumption in a postcode area
grondbeslag_ m^2	Total land take of all houses in the postcode area
Geometry	The geometric representation of the house
Height	The height (in metres) of the house
kWh_2022	Average yearly energy consumption per house in 2022
kWh_connections	Number of power connections in postcode area
kWh_total	Total energy consumption in a postcode area
Net Metering (Ranges from 0 to 100 %)	At 0% all energy is self-generated, and the connection delivers more energy back to the grid than is consumed. 100% indicates only receiving energy
pandcount_2022	Number of buildings in postcode area
pc6	Post Codes
lat	the latitude of the postcode area
lon	the longitude of the postcode area
Volume	The volume of the house (area multiplied by the height)
volume_m^3_p6	Total building volume in a postcode area

APPENDIX B

This Appendix describes the spread of the data for certain features with the use of box plots, and histograms. The box plots provide insights in the spread of a specific feature. They show the quartiles, and the median of the data. Moreover, any outliers in the data are marked by individual dots, making them easily detectable. The distributions represent the frequency of occurring values within different intervals.

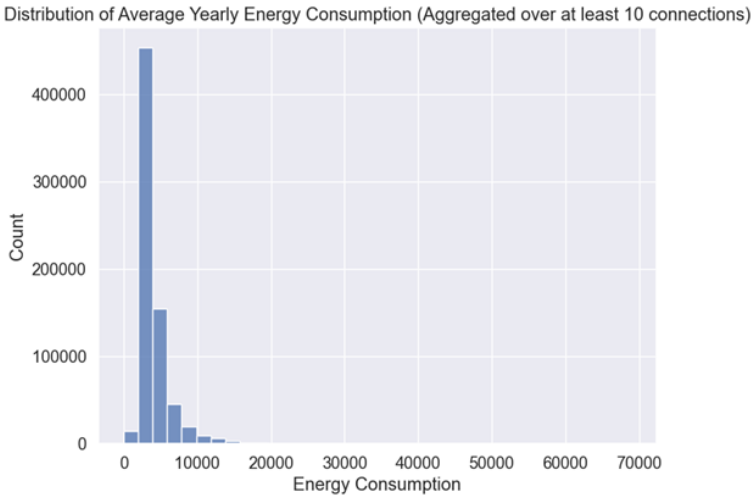

Figure 17: Distribution - Energy Consumption

The distribution of the Energy consumption feature in Figure 17 indicates that there are a substantial number of outliers, which is also reflected in the box plot in Figure 18.

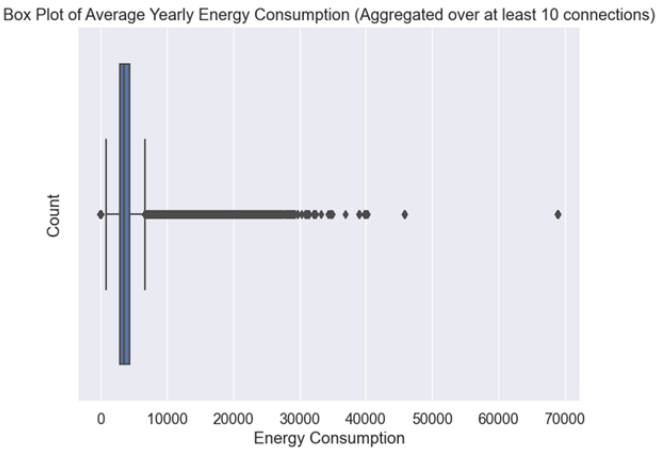

Figure 18: Box Plot - Energy Consumption

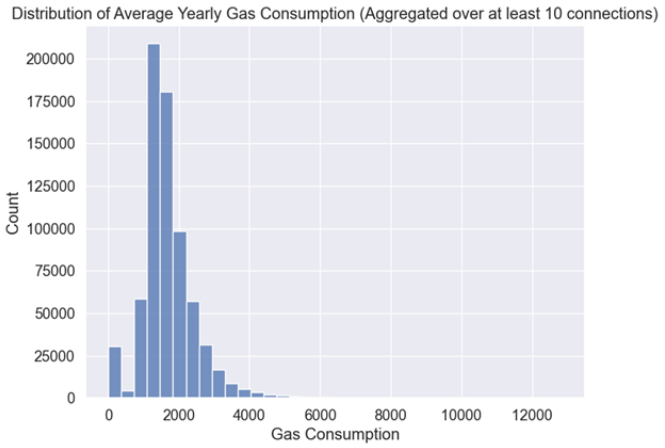

Figure 19: Distribution - Gas Consumption

The distributions and box plots in Figures 19 and 20 showing average gas consumption also demonstrate the presence of numerous outliers. It can be observed that most houses consume around 2,000 cubic gases.

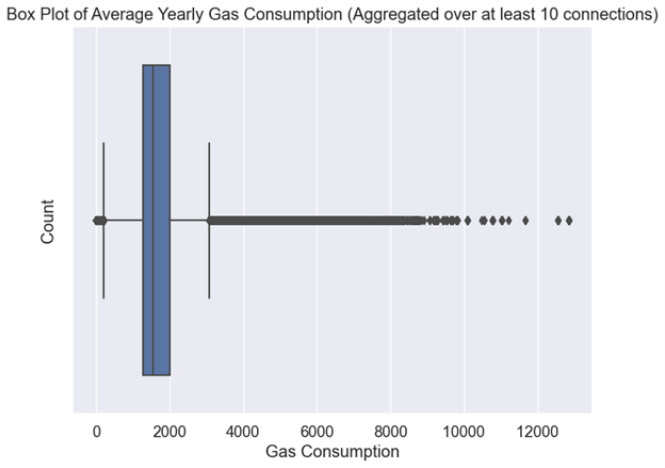

Figure 20: Box Plot - Gas Consumption

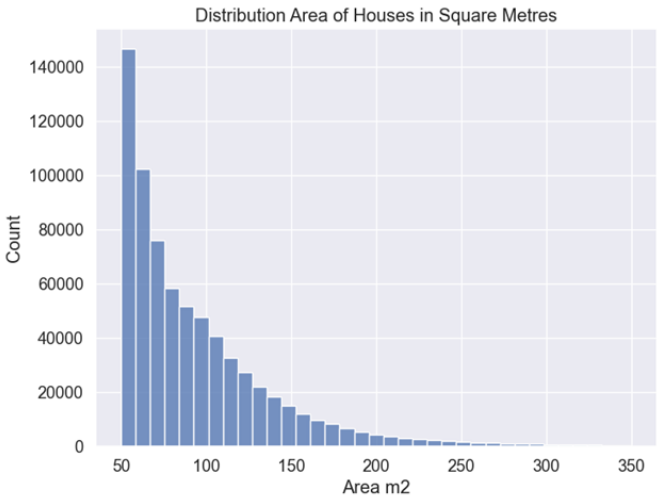

Figure 21: Distribution - Area

The *Area* feature (Figure 21) had a right-skewed distribution, indicating fewer large houses in the dataset. For the *Net Metering* feature (Figure 22), the distribution was left-skewed, meaning that a considerable number of houses delivered a high percentage of energy back to the grid.

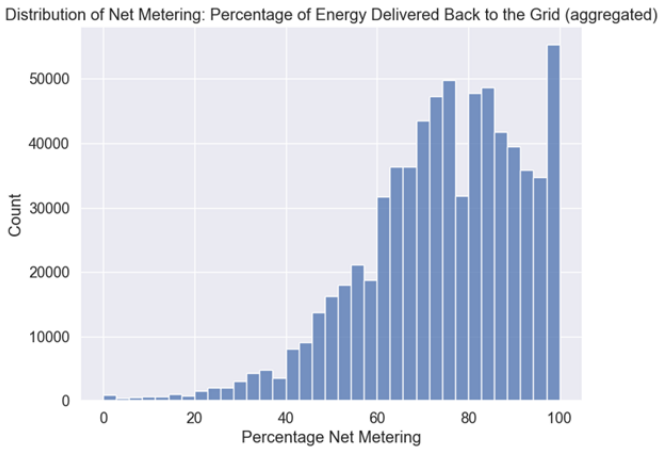

Figure 22: Distribution - Net Metering

As can be seen from Figure 23, the *construction year* feature was extremely right-skewed. This indicated that there were more houses built after the year 1900. There were, naturally, houses that have been in existence for a longer time, and these were therefore not considered as outliers.

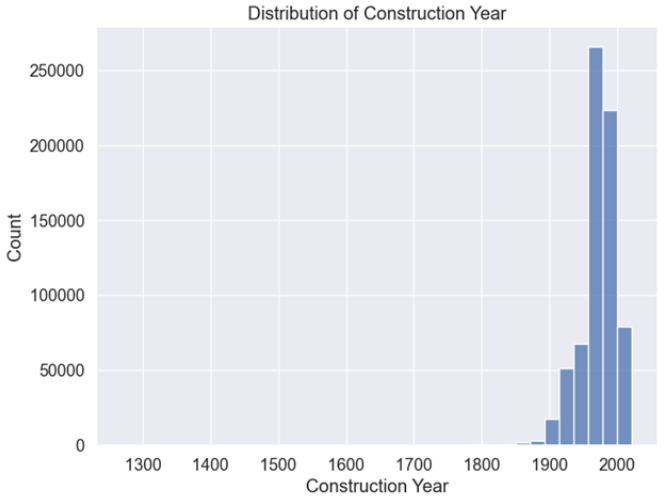

Figure 23: Distribution - Construction Year

The distribution of the *height* was slightly normally distributed. There was, however, a short peek in houses with heights of 3 metres, which were probably bungalows.

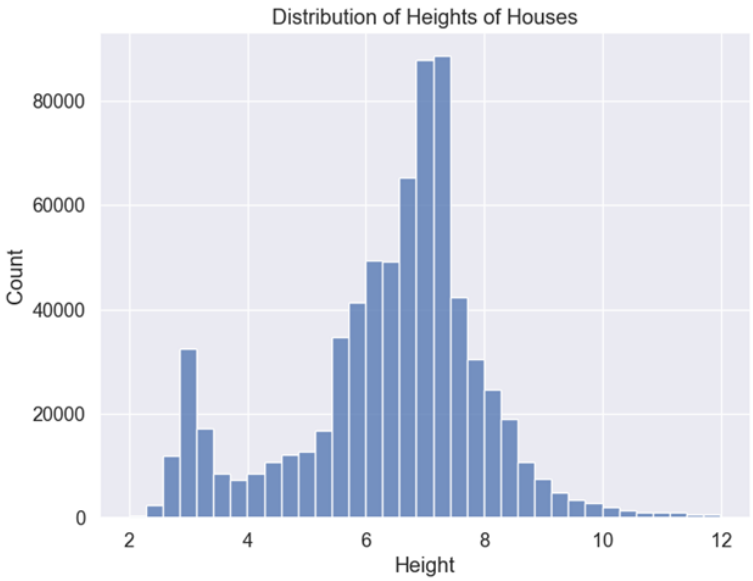

Figure 24: Distribution - Height

Figure 25 depicts the distributions of the *Area*, *Height*, *Construction Year*, *Net Metering*, and *gas*, and *energy* features after the removal of extreme outliers. The *gas* and *energy* distributions were more visible. It can be seen that they were now more normally distributed.

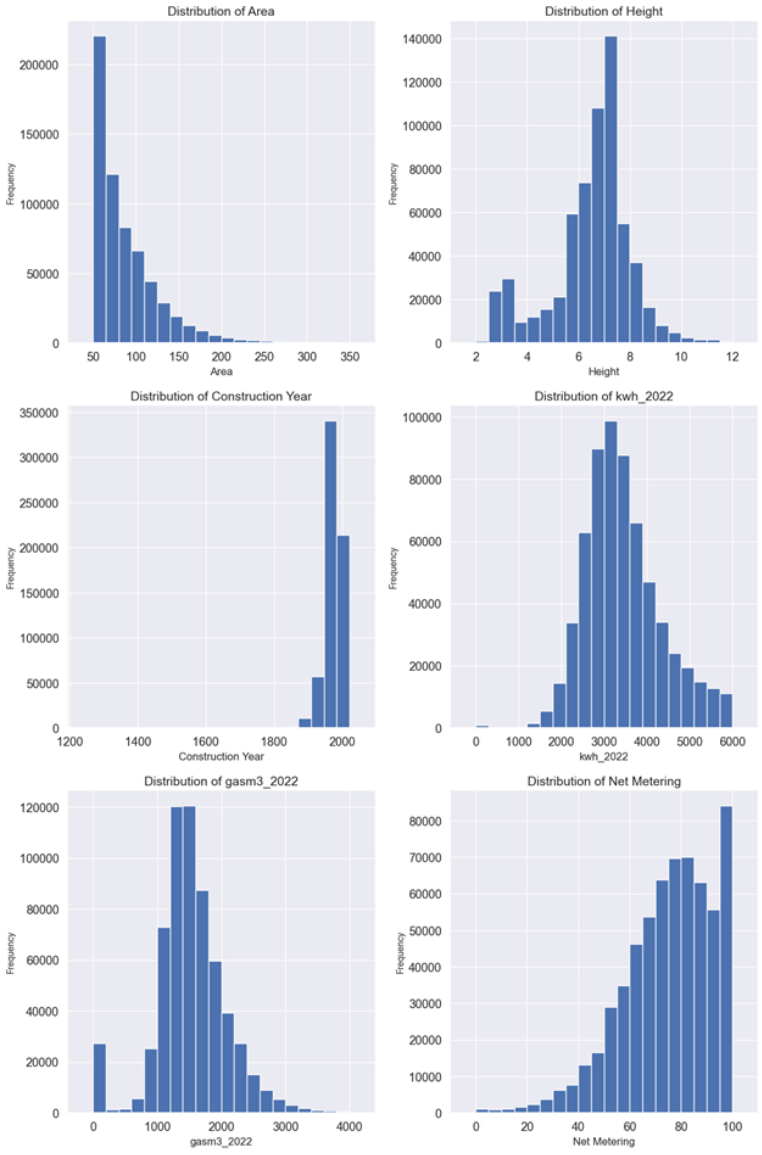

Figure 25: Distributions after Removal of Outliers

APPENDIX C

Random Forest Algorithm

The base component of the Random Forest algorithm is the well-known Decision Tree (DT) algorithm. The DT has been appreciated for its interpretability and explainability due to its intuitive and visual representation of the decision-making process, which is explained in the following paragraphs.

As shown in Figure 26, DTs are composed of nodes and branches. The tree starts at the top with a root node, where a question is asked about a feature of the input data. The branches then split into the possible outcomes of the question. After the split, new questions (child nodes) can be asked about the subsets of the data. This process repeats itself until a stopping criterion is encountered. Finally, after all the questions have been asked, there is a leaf node to make the final decision, prediction or classification.

How does a DT decide on which questions to ask? It relies on the concept of the so-called *information gain*. The information gain is a method that evaluates how much useful information a certain question provides. The aim is to choose the questions that provide the most information (the highest *information gain*) about the target classes (in this study, the energy labels). This process will eventually result in the most informative splits of the data.

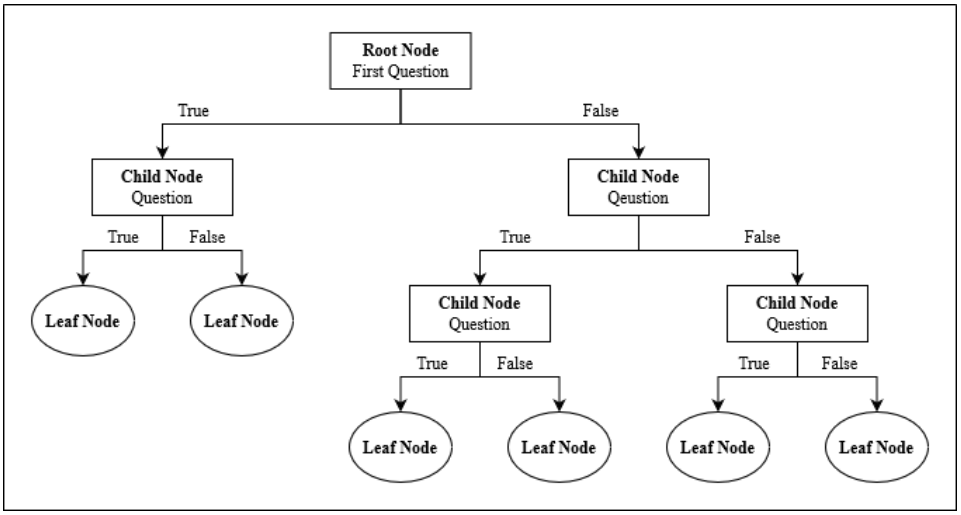

Figure 26: Decision Tree

A Random Forest is essentially a collection of a multitude of individual Decision Trees. Each DT in the forest individually provides a class prediction. For instance, there are a hundred trees in the forest, which

all bring out a prediction. The forest will count the number of trees that predict a certain class. If the majority (>50) of trees predict, for instance, class A, then that is the final prediction. This approach uses the collective knowledge of the individual trees to ensure a more reliable prediction and at the same time reduce the overfitting problem of a single DT (Breiman, 2001).

The "Random" in Random Forest comes from the concept of bagging. Each individual Decision Tree is trained on a resampled training set. This training set is the result of random resampling, with replacement, from the original dataset. The replacement ensures that each training set is unique, and hence all classifiers "learn" on a separate set. If all training sets would be identical, then stacking classifiers makes no sense (Maclin & Opitz, 1997).

APPENDIX D

The data cleaning, preprocessing, and modelling was performed in Python Version 3.9.13 The packages and libraries used were:

- Pandas (development team, 2020)
- Geopandas (Jordahl et al., 2020)
- Numpy (Harris et al., 2020)
- Matplotlib (Hunter, 2007)
- Seaborn (Waskom, 2021)
- Scikit-learn (Pedregosa et al., 2013)
  - RandomForestClassifier
  - imbalanced-learn
  - scikit-learn semi-supervised
- Pytorch TabNet (TabNetClassifier) (Aghajanzadeh, Sercan, 2019)
- xgboost (XGBClassifier) (Chen & Guestrin, 2016a)
- Shapely (Gillies et al., 2007)
- Optuna (Akiba et al., 2019a)

In addition, the open source Geographic Information System QGIS version 3.22.14 was used to analyse some test results (QGIS Development Team, 2009).

APPENDIX E: RANDOM FOREST RESULTS

This appendix presents the results of the experiments conducted with the Random Forest model. First, the confusion matrices are shown for the baseline RF, the tuned RF, and tuned RF with SMOTE-Tomek. Secondly, the performances per class per classifier (baseline, tuned, tuned + SMOTE-Tomek) are presented. Lastly, the Optuna optimization history plots are depicted.

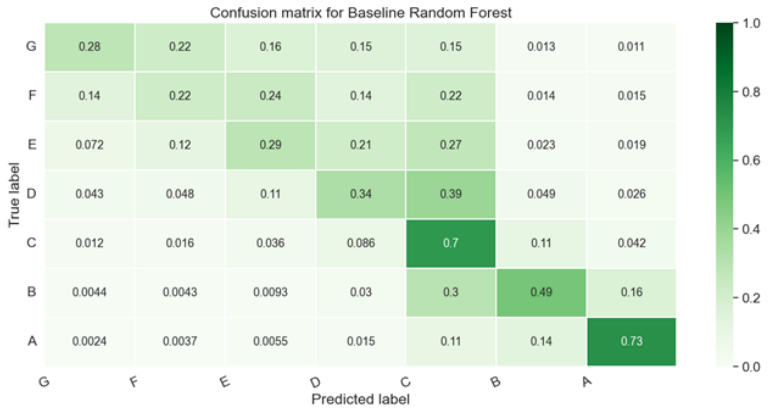

Figure 27: Confusion Matrix: Baseline Random Forest

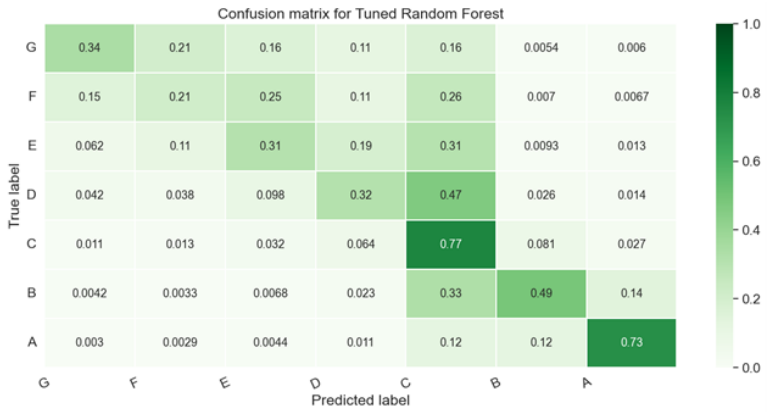

Figure 28: Confusion Matrix: Tuned Random Forest

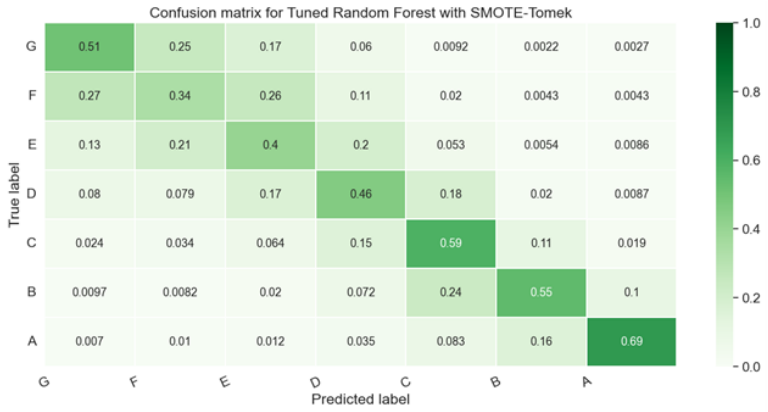

Figure 29: Confusion Matrix: Tuned Random Forest with SMOTE-Tomek

Table 12: Performance per Class: Baseline Random Forest

Class	Precision	Recall	F1-Score	Support
G	0.29	0.28	0.29	1846
F	0.26	0.22	0.24	2555
E	0.31	0.29	0.30	4069
D	0.39	0.34	0.36	7240
C	0.58	0.70	0.63	20175
B	0.55	0.49	0.52	12088
A	0.78	0.73	0.75	15182

Table 13: Performance per Class: Tuned Random Forest

Class	Precision	Recall	F1-Score	Support
G	0.33	0.34	0.34	1846
F	0.28	0.21	0.24	2555
E	0.34	0.31	0.33	4069
D	0.43	0.32	0.37	7240
C	0.58	0.77	0.66	20175
B	0.61	0.49	0.54	12088
A	0.82	0.73	0.77	15182

Table 14: Performance per Class: Tuned Random Forest with SMOTE-Tomek

Class	Precision	Recall	F1-Score	Support
G	0.27	0.51	0.35	1846
F	0.23	0.34	0.28	2555
E	0.30	0.40	0.34	4069
D	0.37	0.46	0.41	7240
C	0.68	0.59	0.63	20175
B	0.58	0.55	0.56	12088
A	0.86	0.69	0.77	15182

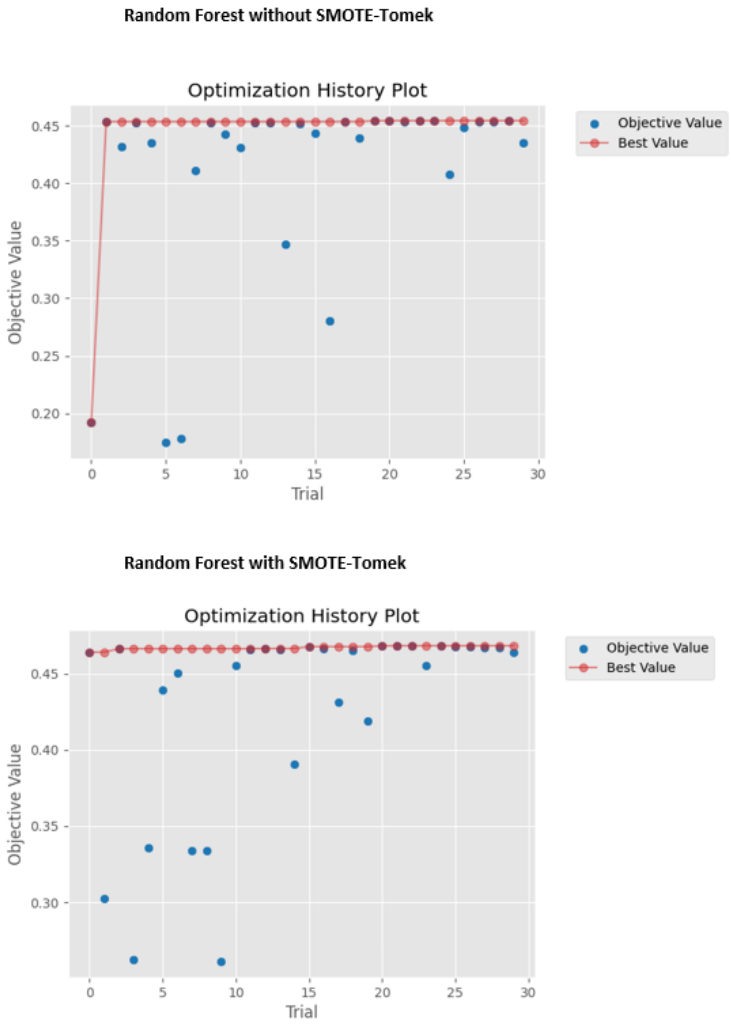

Figure 30: Optuna Optimization History Random Forest

APPENDIX F: XGBOOST RESULTS

This appendix contains the results of the experiments conducted with the XGBoost algorithm. The confusion matrices, performances per class, and the Optuna optimization plots are shown for the baseline XGB, the tuned XGB and the tuned XGB with SMOTE-Tomek.

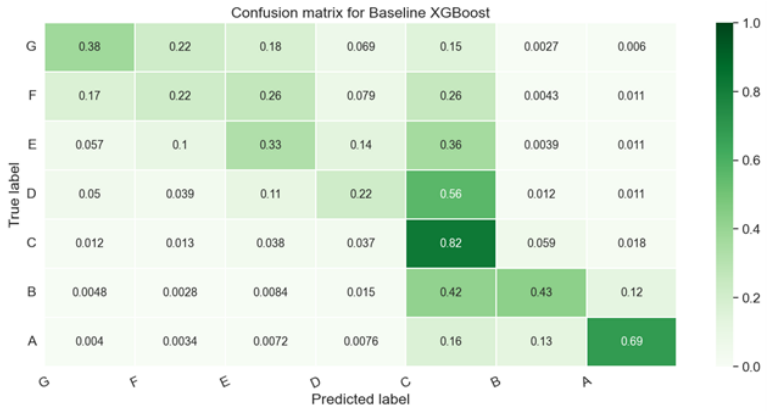

Figure 31: Confusion Matrix: Baseline XGBoost

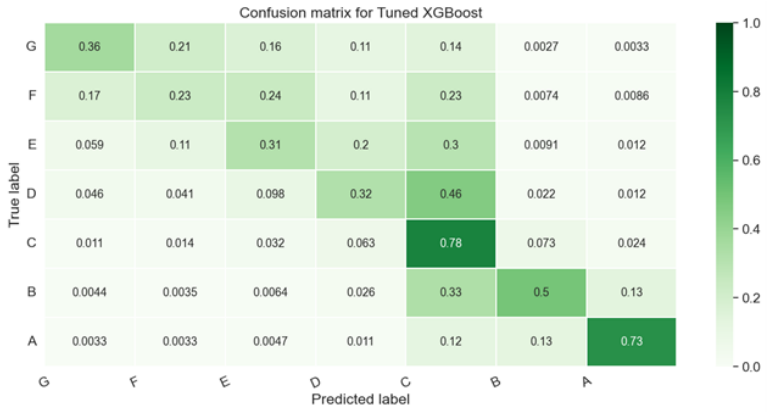

Figure 32: Confusion Matrix: Tuned XGBoost

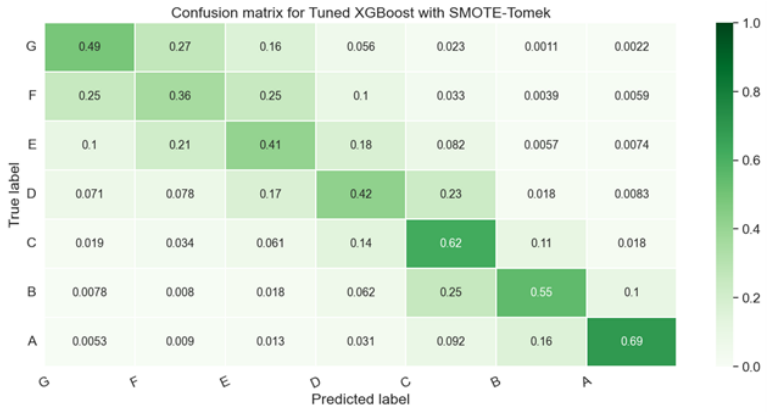

Figure 33: Confusion Matrix: Tuned XGBoost with SMOTE-Tomek

Table 15: Performance per Class: Baseline XGBoost

Class	Precision	Recall	F1-Score	Support
G	0.33	0.38	0.35	1846
F	0.27	0.22	0.24	2555
E	0.32	0.33	0.33	4069
D	0.45	0.22	0.29	7240
C	0.54	0.82	0.65	20175
B	0.62	0.43	0.51	12088
A	0.84	0.69	0.76	15182

Table 16: Performance per Class: Tuned XGBoost

Class	Precision	Recall	F1-Score	Support
G	0.33	0.36	0.35	1846
F	0.28	0.23	0.26	2555
E	0.34	0.31	0.33	4069
D	0.43	0.32	0.37	7240
C	0.58	0.78	0.67	20175
B	0.62	0.50	0.55	12088
A	0.83	0.73	0.78	15182

Table 17: Performance per Class: Tuned XGBoost with SMOTE-Tomek

Class	Precision	Recall	F1-Score	Support
G	0.30	0.49	0.37	1846
F	0.24	0.36	0.29	2555
E	0.30	0.41	0.35	4069
D	0.38	0.42	0.40	7240
C	0.66	0.62	0.64	20175
B	0.58	0.55	0.57	12088
A	0.86	0.69	0.77	15182

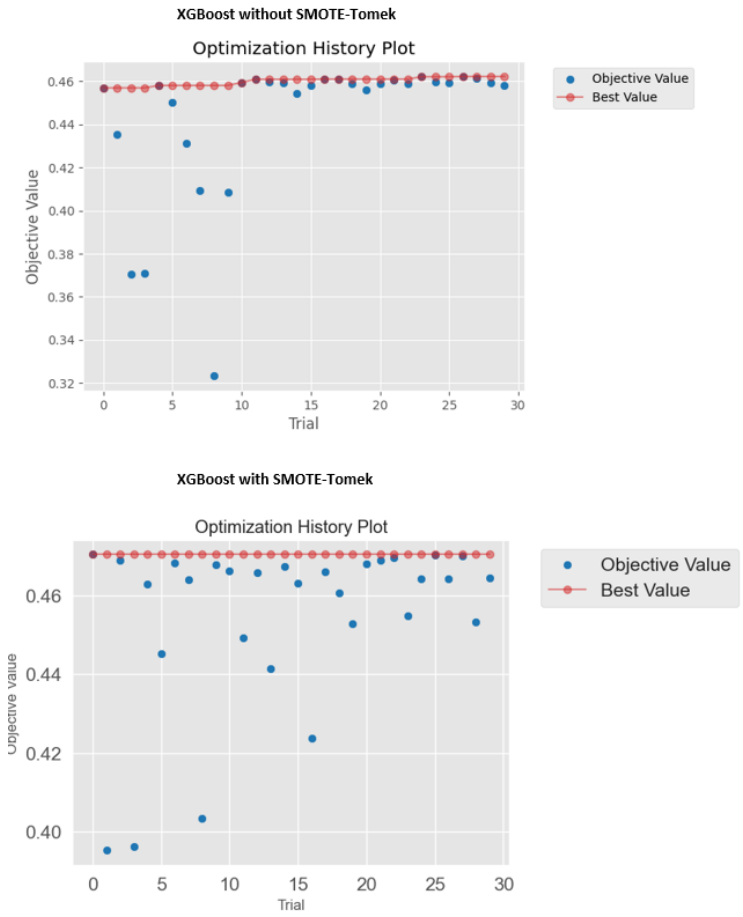

Figure 34: Optuna Optimization History XGBoost

APPENDIX G: TABNET RESULTS

This appendix likewise shows the results obtained from the experiments with TabNet. Once more, confusion matrices, performance by class and graphs of the Optuna optimisation history are presented.

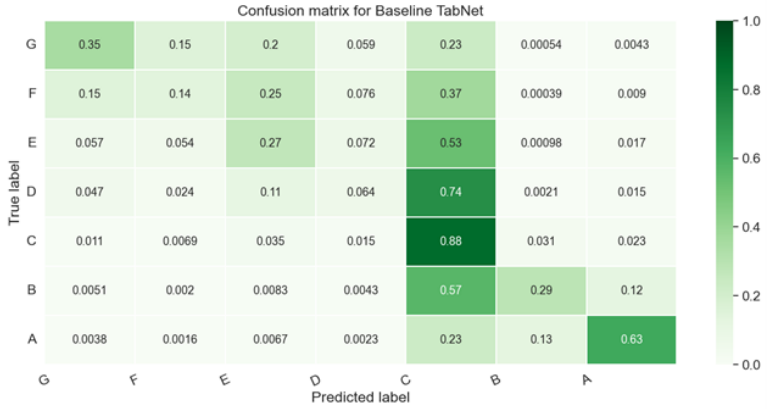

Figure 35: Confusion Matrix: Baseline TabNet

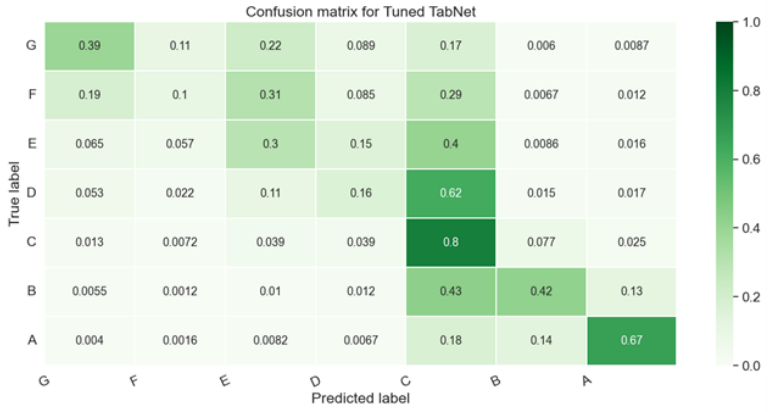

Figure 36: Confusion Matrix: Tuned TabNet

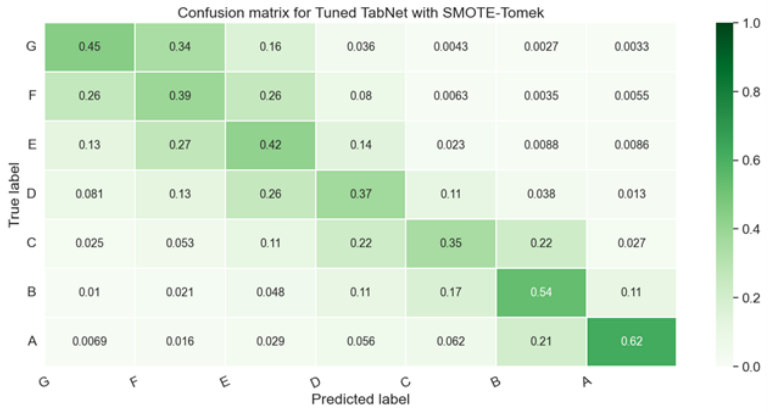

Figure 37: Confusion Matrix: Tuned TabNet with SMOTE-Tomek

Table 18: Performance per Class: Baseline TabNet

Class	Precision	Recall	F1-Score	Support
G	0.33	0.35	0.34	1846
F	0.29	0.14	0.19	2555
E	0.29	0.27	0.28	4069
D	0.32	0.06	0.11	7240
C	0.48	0.88	0.62	20175
B	0.58	0.29	0.38	12088
A	0.82	0.63	0.71	15182

Table 19: Performance per Class: Tuned TabNet

Class	Precision	Recall	F1-Score	Support
G	0.32	0.39	0.35	1846
F	0.26	0.10	0.15	2555
E	0.28	0.30	0.29	4069
D	0.37	0.16	0.23	7240
C	0.52	0.80	0.63	20175
B	0.57	0.42	0.48	12088
A	0.82	0.67	0.73	15182

Table 20: Performance per Class: Tuned TabNet with SMOTE-Tomek

Class	Precision	Recall	F1-Score	Support
G	0.25	0.45	0.32	1846
F	0.19	0.39	0.26	2555
E	0.22	0.42	0.29	4069
D	0.27	0.37	0.31	7240
C	0.64	0.35	0.46	20175
B	0.45	0.54	0.49	12088
A	0.83	0.62	0.71	15182

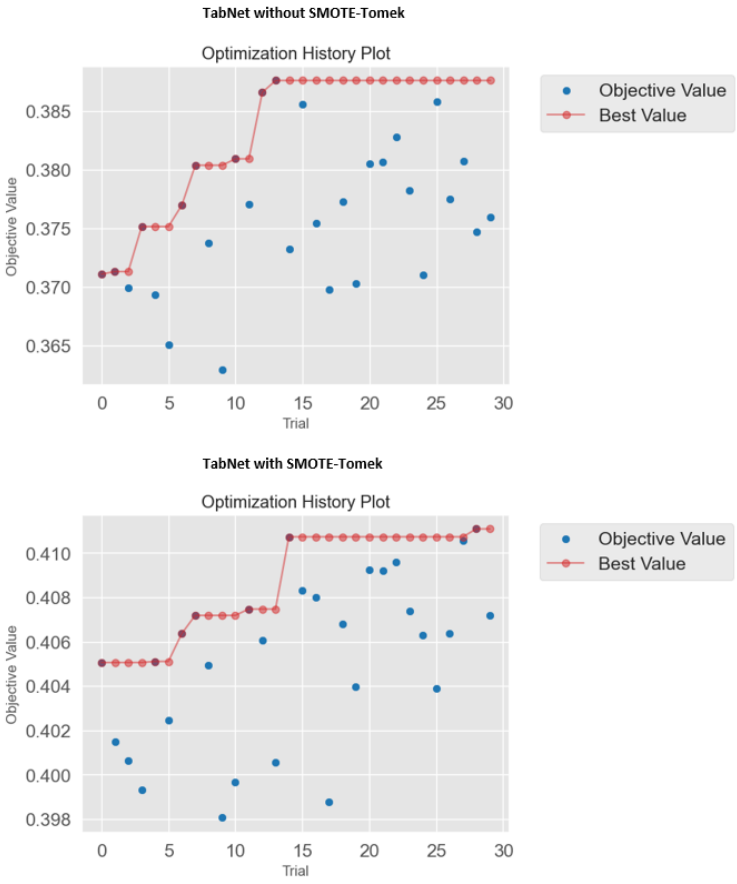

Figure 38: Optuna Optimization History TabNet

APPENDIX H: TRUE VERSUS PREDICTED ENERGY LABELS

Figure 39: Aggregated issue: True versus Predicted Energy Labels

Figure 40: Aggregated issue: True versus Predicted Energy Labels

Figure 41: Confusion issue: True versus Predicted Energy Labels

APPENDIX I: DUTCH DATASET RESULTS

This appendix shows results of the RF, and XGBoost tuned on the dataset of the Netherlands without SMOTE-Tomek applied to the training data. Due to time constraints, this dataset was not used for the thesis. Instead, the smaller, but big enough Brabant dataset was used. Table 21 reveals that the size of the data did not greatly impact the performances of the RF, and XGBoost classifiers.

Table 21: Performance of Classifiers Tuned with the Dataset of the Netherlands or the dataset of Brabant

Model	**Dataset**	**Precision**	**Recall**	**F1-Score**	**Balanced Acc.**	**Cohen's Kappa**
Tuned RF	Netherlands	0.485	0.472	0.471	0.472	0.476
Tuned RF	Brabant	0.484	0.454	0.464	0.454	0.467
Tuned XGBoost	Netherlands	0.488	0.472	0.472	0.472	0.478
Tuned XGBoost	Brabant	0.490	0.463	0.471	0.463	0.475