UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Universität Paderborn
Fakultät für Wirtschaftswissenschaften
Department Wirtschaftsinformatik
Lehrstuhl für Wirtschaftsinformatik, insb. Data Analytics

Studienarbeit Predictive Analytics

ASHRAE Great Energy Predictor III

by
Raphael Reimann (7158324)
reimannr@mail.uni-paderborn.de

Supervisor: Prof. Dr. Oliver Müller

Submitted on 7th March 2022

Contents

List of Figures

List of Tables

1 Business Understanding

1.1 Problem

The building sector is responsible for about 30% of the global energy usage in 2013 with demands rising in the following years (?, ?). Statistical analysis of buildings energy use can play a significant role in lowering a buildings energy usage, its carbon emission when it is relying on fossil fuels for power, and lower the overall cost of operation for businesses and private households (Miller, Kathirgamanathan et al., 2020). However, due to the great complexity of energy systems in buildings and the varying energy and building types, the analysis of energy usage is a complex problem. In the literature the main types of energy explored is healing/cooling load, hot water and energy consumption. The building types explored vary across residential, office, industrial, and education buildings (Zhao, Magoulès & Magoulès, 2012). The research community has commonly looked at open datasets in conjunction with competitions to predict building energy usage over time, given a set of input features of the building. Several open datasets covering building energy data have been released within the scientific community so far covering different building types, such as office and residential, and different energy systems such as ventilation systems and heat pumps.

One of those datasets has been released with the competition ASHRAE Great Energy Predictor III held in 2019. The competition as it is presented on Kaggle is accompanied by a publication (Miller, 2019) which outlines why the effort of building energy prediction is necessary. The field of predicting the energy usage of buildings can be divided into short-term and long-term prediction. Short-term prediction seeks to assist real-time control of heating and cooling systems as well as scheduling of power stations at a timescale of hours or days. The objective of long-term prediction is to be able to do What-If analyses for energy conservation measures. A baseline model can predict how much energy a building would use in a certain setting based on historical data if a energy conservation measure had not been enacted. The difference between the predicted usage based on historical data and the actual usage is the saving that can be attributed to the implementation of the energy conservation measure. This is crucial for builidng owners to evaluate the impact of such a measure on energy savings and related emissions and costs. Cheng Fan et al. (2020) found that data-driven methods are a cost-effective way to analyze retrofitting measures towards their potential to lower energy usage for a large number of buildings. Geyer, Schlüter and Cisar (2017) analyzed how clustering methods can be used for retrofitting strategies and the sensitivity of different buildings towards those measures. Marasco and Kontokosta (2016) used a machine learning classifier to evaluate the savings from different energy conservation measures for buildings in New York City.

1.2 Literature Review

We conducted a comprehensive literature search on academic search platforms Google scholar and Science Direct using the query string ¨`(data driven OR machine learning OR forecasting) AND (building energy OR smart building OR energy management OR retrofit)`¨. We then identified articles with a related topic to this paper and conducted forward and backward searches. Reviewing the current literature, the majority of articles focus on data-driven modeling of the building performance. The majority of studies have followed a typical data analysis framework, which consists of several phases. First during data-preprocessing the raw data is transformed into useful information for predictive modeling. Typical methods applied in the preprocessing stage are data cleaning, reduction, transformation, and partitioning. Second is the modelling of machine learning algorithms to create a model which accurately displays the relationships in the underlying data. In the next phase the results of the modeling phase are evaluated according to their predictive accuracy. In a final step the model is applied to help actual decision making of building energy management tasks, such as fault detection and optimal control of HVEC systems. Amber et al. (2017) used multiple linear regression to predict the daily electricity consumption for a commercial and an building of the education sector. They collected five years of data and trained their model on four years. Input features were air temperature, relative humidity, solar radiation, wind speed, weekday, building type all collected on a daily mean level. Pulido-Arcas, Pérez-Fargallo and Rubio-Bellido (2016) used a multiple linear regression model to predict total energy consumption, i.e. electricity and gas, in office buildings in Chile. They trained individual models for nine different climatic locations achieving a mean absolute error between 0.11 kW and 0.41 kW in their forecasting. Tso and Yau (2007) compared different modeling techniques, i.e. multi-layer perceptron (MLP), stepwise regression and Decision tree models, to predict electricity consumption in residential buildings in Hong-Kong. They found that decision tree model outperformed the regression and the MLP model in the summer time, and the MLP achieved a higher accuracy in the winter time. Yu, Haghighat, Fung and Yoshino (2010) used a decision tree model to classify the energy usage of residential buildings in Japan. Using input features such as air temperature, building characteristics, and occupant number they classified the energy usage intensity as either high or low, achieving a 92% success rate in classification. Wang, Wang, Zeng, Srinivasan and Ahrentzen (2018) compared Random forest to a decision tree model and a Support Vector Regression (SVR) model to predict electricity consumption of different non-residential buildings. They collected data over a year and trained their model on 80% of that data. Input features they used were air temperature, pressure, precipitation, estimated occupancy, time, and workday yes/no. The random forest model performed better than the decision tree and the Support vector regression model. Papadopoulos, Azar, Woon and Kontokosta (2018) compared different ensemble models based on Decision trees as base learners all implemented with the Scikit-Learn Python library. Specifically they implemented a Random Forest model, , an extra randomized tree model and a Gradient Boosting Decision Tree model and compared their performance on an benchmarking dataset with results from previous studies that were also using that dataset (Tsanas & Xifara, 2012). They found that especially the GBDT model improved the forecasting accuracy compared to previous studies that focused on regression and Random forest models or Support Vector machines. Edwards, New and Parker (2012) apply Regression, Feed-Forward Neural Network, Support Vector Regression, Least Squares Support Vector Machine and additional NN-based methods on a residential data set containing sensor measurements collected every 15 minutes to predict next hour residential building consumption. They conclude that Neural Network based methods perform best on commercial buildings but poorly on residential data. Least Squares Support Vector Machines perform best on residential data. Burak Gunay, Shen, Newsham and Ashouri (2019) compared different modeling techniques, i.e. change-point, random forest, and artificial neural networks, to identify several types of energy use anomalies. The used a dataset from 35 Canadian office buildings with primarily weather data as features. Roth, Chadalawada, Jain and Miller (2021) use sub-hour smart meter data streams to forecast building-level electricity demand. They use Bayesian Structural Time Series for probabilistic load forecasting and show that it outperforms traditional ARIMA time series forecasting. They also identify possible application areas such as Load Forecasting for Apartment-level hourly loads, Submeter load forecasting and segmentation and measument and verification for behavioral demand response. Zhao et al. (2012) cover methods to predict building energy usage with statistical and engineering methods. They concluded that especially support vector machines (SVMs) are able to predict energy usage solely based on past energy data, if metadata on the building type are not available.

1.2.1 Competition solution overview

While the goal of this paper is to grasp the essence of the data set thoroughly and explore different pre-processing and modeling techniques and not to beat the best solutions from the initial Kaggle competition, it is helpful to look at the approaches that worked well on this data set.

The fifth-place solution was submitted by Tatsuya Sano, Minoru Tomioka, and Yuta Kobayashi, who are all students at the University of Tsukuba in Tsukuba, Japan. Their approach explored different data preprocessing techniques and compared percentile based and proportion based target variables. They used LightGBM models exclusively and tuned the number of trees hyperparameter on a seperate model for each building before ensembling them with respect to the building occurence. (Miller, Arjunan et al., 2020)

The fourth-place solution from Jun Yang of the University of China in Chengdu used time-based features and lag features from the weather data, i.e. using aggregated data from a timeframe lagging behind the respective data point. This approach used XGBoost and LightGBM models that were ensembled with information retrieved from data that was leaked during the competition.

The third-place solution by Xavier Capdepon used a log transformation of the target variable before training. Extensive weather features were extracted and lag features were also used to train several models including Catboos, Neural Networks, and LightGBM, which were then ensembled.

The second-place solution was submitted by Rohan Rao, Anton Isakin, Yangguang Zang, and Oleg Knaub and focused on the manual removal of outliers in the data. They trained different models including XGBoost, LightGBM, Catboost, and Feed-forward Neural Networks on different subsets of the training data and ensembled them.

The first-place solution by Matthew Motoki and Isamu Yamashita focused on the removal of data anomalies and the imputation of missing values in the weather data as well as the correction of time zones. They extracted additional features including temporal information such as holidays or time of day. They split the training data into different subsets based on the meter type, the primary use of the building and the site and trained different models based on Catboost, LightGBM, and Multi-Layer Perceptrons architectures. The predictions of the individual models were then combined with a weighted mean approach to predict the target values of the test set.

Reference	Dataset	Features	Method	Timeframe
(Xuemei, Jin-hu, Lixing, Gang & Jibin, 2009)	Simulated	Air temperature, relative humidity, solar radiation	Least-Square-Support-Vector-Machine, ANN	4 months
(Fan, Xiao & Wang, 2014)	Real	Air temperature, dew point temperature, relative humidity, pressure, cloud coverage, rainfall, hours of reduced visibility, solar radiation, evaporation, wind speed	MLR, ARIMA, SVM, DT, MLP, Multivariate adaptive regression splines, K-nearest-neighbour	One year
(Burak Gunay et al., 2019)	Real	Air temperature, wind speed, horizontal solar irradiance, binary work hours indicator	Change-Point, RF, ANN	
(Roth et al., 2021)	Real	Electricity use of various appliances, i.e. air conditioning, water heater	Bayesian Structural Time Series	1.5 years
(Amber et al., 2017)	Real	Air temperature, relative humidity, solar radiation, wind speed, weekday, building type	MLR	Five years
(Pulido-Arcas et al., 2016)	Real	number of floors, square footage, form ratio, wall-to-window ratio, energy efficiency ratio, heating and cooling emission factors	MLR, ANN	Two years
(Tso & Yau, 2007)	Real	power rating appliances, winter/summer, housing type, household characteristics	DT, MLP, MLR	Two years
(Yu et al., 2010)	Real	air temperature, building characteristics, occupant number, energy saving measure	DT (classification)	Two years
(Wang et al., 2018)	Real	Air Temperature, Dew point, humidity, pressure, wind speed, solar radiation	RF, DT, SVR	One year
(Papadopoulos et al., 2018)	Benchmark (Simulated)	Relative compactness, suface area, wall area roof area, overall height, orientation	RF, Extra, randomized trees, GBDT	
(Edwards et al., 2012)	Benchmark (ASHRAE dataset)	Temperature, solar flux, sin of current hour, cosine of current hour	-	6 months
(Karatasou, Santamouris & Geros, 2006)	Benchmark (ASHRAE dataset)	Temperature, solar flux, humidity, wind speed, date, sin and cos of current day of week, hour of day, day of year	ANN (Feed-forward Neural Network)	6 months
(González & Zamarreño, 2005)	Benchmark (ASHRAE dataset)	Current and forecasted temperature, date	ANN (Feedback)	6 months
(Borges, Penya, Fernández, Prieto & Bretos, 2013)	Real & Benchmark (ASHRAE)	-	SVM, Autoregressive	18 months & 6 months

Table 1.1: Overview of related literature

2 Data Understanding

The underlying component of all data-driven approaches is the data and its quality. The data used in existing research in building energy performance can be classified into two main categories, i.e. measured data and simulated data. Measured data is commonly gathered from experiments and on-site measurement devices, that include building automation systems, energy meters, weather stations, and Internet-of-Things sensors (IoT). Data retrieved from this class of sources can reveal real operational conditions in which a buildings energy system operates. The data quality however can be low, because data measured from real sensors is subject to noise, sensor faults, or inaccurate calibration. Simulated data is collected from physics-based models of buildings that can be real or virtual. It is not retrieved from real sensors, which ensures a higher data quality, because it is not subjected to measurement errors. The accuracy of models built based on simulated data is however oftentimes not applicable to usage in the real world, because it does represent actual building operations and cannot handle sensor problems that occur in real-world usage.

Common datasets that are publicly available and have been used for building energy research in the past include the ASHRAE's Great Building Energy Predictor Shootout (Karatasou et al., 2006) and the UCI machine learning repository (Marino, Amarasinghe & Manic, 2016).

2.1 Data Sources

The dataset used in this paper is the ASHRAE Great Energy Predictor III Kaggle competition dataset, which is taken from (Miller, Kathirgamanathan et al., 2020). The data collection process by Miller and the ASHRAE Team started in March 2018. They state that the goal in the data collection phase was to create the largest and most diverse dataset possible to challenge the contestants to create the most generalizable models for the benefit of the energy prediction research community. The data was collected from both publicly available sources and closed systems where data was extracted together with the respective facility management teams. The initial dataset included 61,910,200 energy measurements taken from 16 sites worldwide within a timespan between January 1, 2016 and December 31, 2018. Miller notes that a majority of data is taken from university buildings, which is why the primary use of the buildings is education. An overview of the data and variables is presented in Table 2.1.

File Name and Description	File Variables and Short Description
`train.csv` - This file includes one year of hourly time-series (8760 samples per meter) for each meter. The date range for this training set is Jan 1 - Dec 31, 2016. These data are used to train the prediction models.	`building_id` - Foreign key for the building metadata. `meter` - The meter id code. Read as 0: electricity, 1: chilledwater, 2: steam, 3: hotwater. Not every building has all meter types. `timestamp` - When the measurement was taken. `meter reading` - The target variable. Energy consumption in kWh (or equivalent). Note that this is real data with measurement error, which we expect will impose a baseline level of modeling error.
`building_meta.csv` - This file includes the characteristic data from each building in the competition	`site_id` - Foreign key to match with weather.csv. `building_id` - Foreign key for training.csv. `primary_use` - Indicator of the primary category of activities for the building based on EnergyStar property type definitions. `square_feet` - Gross floor area of the building. `year_built` - Year building was opened. `floor_count` - Number of floors of the building.
`weather_[train/test].csv` - Weather data from a meteorological station as close as possible to the site. The data is for all sites and spans from Jan. 1, 2016 to Dec. 31, 2018.	`site_id` - Foreign key to match with the meta.csv file. `air_temperature` - Degrees Celsius `cloud_coverage` - Portion of the sky covered in clouds, in oktas `dew_temperature` - Degrees Celsius `precip_depth_1_hr` - Precipitation in millimeters `sea_level_pressure` - Millibar/hectopascals `wind_direction` - Compass direction (0-360) `wind_speed` - Meters per second
`test.csv` - The submission files use row numbers for ID codes in order to save space on the file uploads. test.csv has no feature data; it exists so you can get your predictions into the correct order. The test data submissions span from Jan. 1, 2017 to Dec. 31, 2018.	`row_id` - Row id for the submission file `building_id` - Building id code `cloud_coverage` - Portion of the sky covered in clouds, in oktas `dew_temperature` - Degrees Celsius `meter` - The meter id code `timestamp` - Timestamps for the test data period

Table 2.1: Overview of the files provided in the competition

2.2 Exploratory Data Analysis

Missing values

In 2.1 we see that our dataset has missing values across some columns. Columns building_id, meter, timestamp, meter_reading, site_id, primary_use, and square_feet all have no missing values. The features with the highest rate of missing values are floor_count, year_built, and cloud_coverage which we have to look out for in data preparation.

Distribution of dependent variable

We find that the target variable `meter_reading` is heavily skewed with most values being close to 0 kWh but some outliers in the millions (see figure 2.2). This shows us that we have to preprocess the data to remove outliers in our dataset.

Using data analysis to identify outliers, we found that `meter` 2 (Steam) of building 1099 has really high values that can not be explained by variation of the building type or weather circumstances (see 2.3). During the course of the competition it also became apparent that some readings from site 0 (encompassing buildings 1-104) have energy readings using the wrong units. This has been noted by the community on

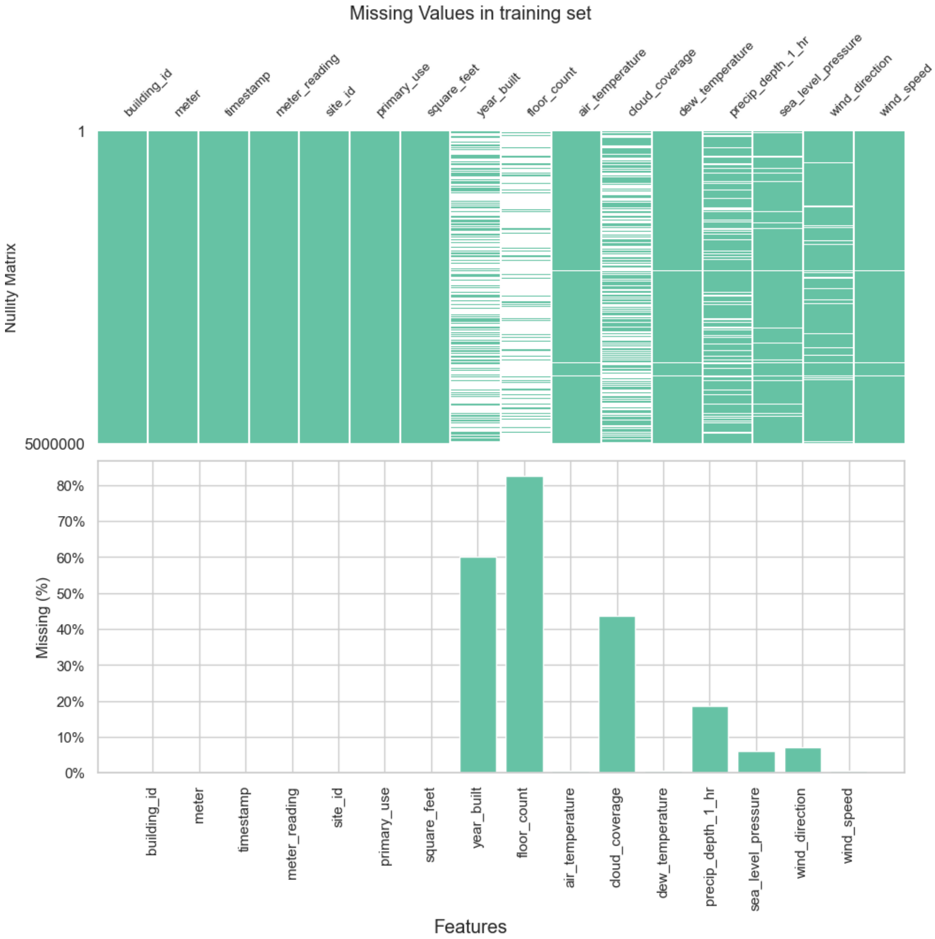


Figure 2.1: Meter reading per site (mean)

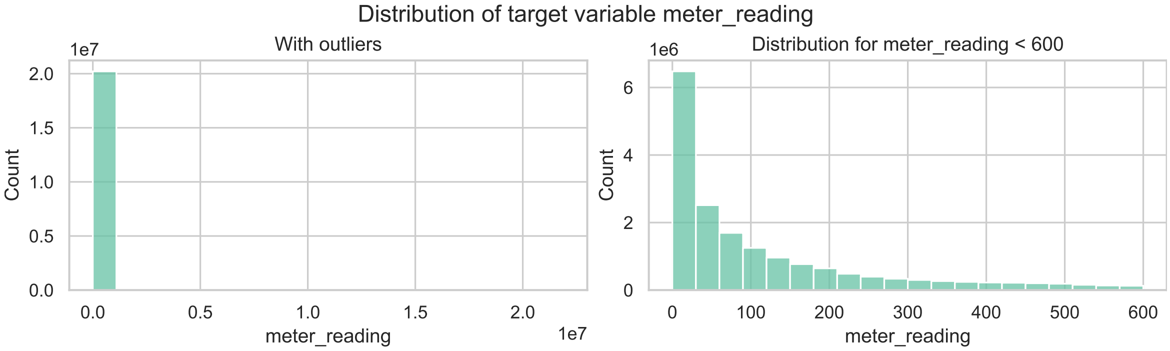


Figure 2.2: Distribution of target variable

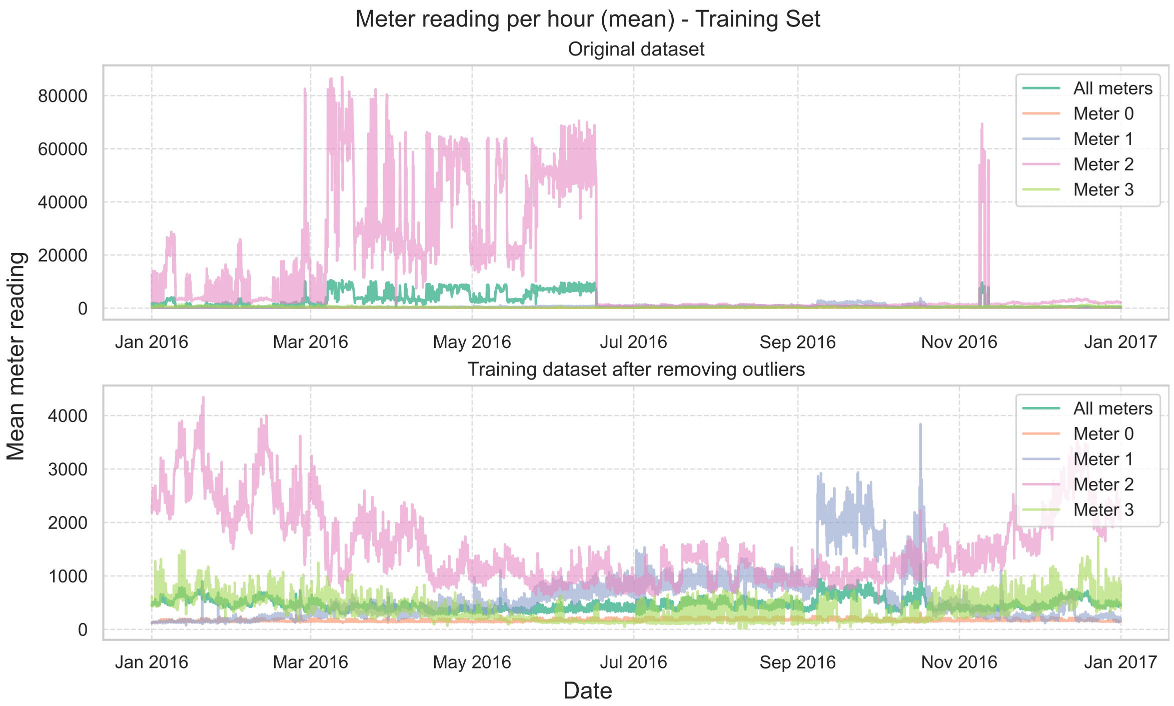


Figure 2.3: Meter reading per site (mean)

Kaggle (Yee, n.d.) and by Miller, Kathirgamanathan et al. (2020). We will continue exploring our data with the outliers removed.

3 Data Preparation

To feed the data into our machine learning models we have to preprocess to handle missing values, apply standardization, and encode non-numerical values. During the training process we have to be careful to not leak data from our training set to preprocessing of our validation sets. To avoid this, we use pipelines provided by the Scikit-Learn library, which ease the process of applying all of our preprocessing steps at once and fit them only to our training dataset. The whole preprocessing process is shown in 3.1.

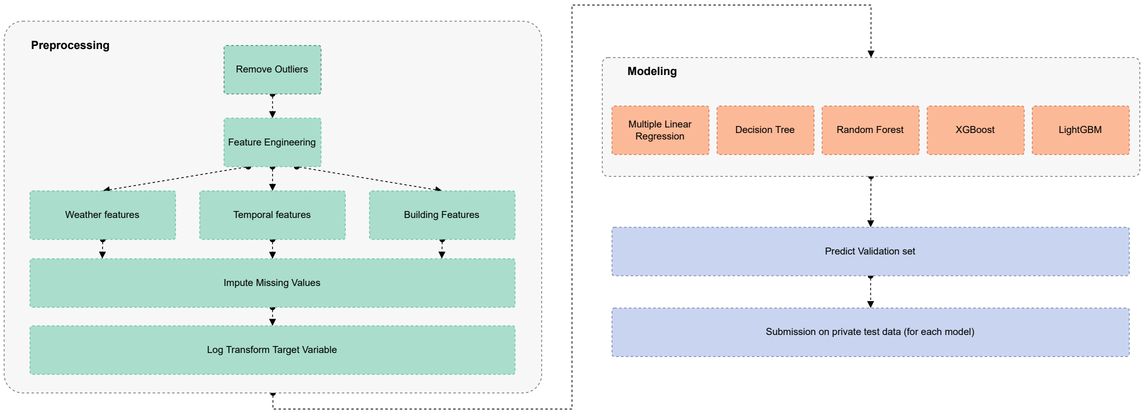


Figure 3.1: Preprocessing flowchart

3.1 Numerical Values

We handle missing values in pipelines for their respective datatype category. We process all features with numerical values by first imputing the missing values with the mean value of the given column. We then standardize the data for each numerical column with $z = \frac{x-\mu}{\sigma}$ where z is the standardized value of x, μ being the mean, and σ being the standard deviation of the column.

Weather Data

During analysis we found that there is a lot of missing weather data. There is data missing for some hours, but also missing values for certain climatological variables at some timestamps, which are Not a Number (NaN) in the dataset. The way the missing data is imputed is heavily inspired from a Kaggle Notebook by user Sandeep Kumar. Here the missing hourly readings are first added to the weather dataframe. The data is then grouped by `site_id` and the day of the observation and missing values for climatological variables `air_temperature`, `cloud_coverage`, `dew_temperature`, `sea_level`, `wind_direction`, and `precip_depth_1_hr` are filled by propagating the last valid observation forward to next valid observation.

Target Variable

During our modeling phase, we have found that training our models on a log-transformed version of the target variable (`meter_reading`) improves our prediction. We use Scikit-Learns `TransformedTargetRegressor` within our pipeline to ensure that we apply $log(1 + x)$ during fitting and $exp(x) - 1$ during prediction.

3.2 Categorical Values

The missing values of features with categorical values are handled by imputing them with the most frequent value of the column, as the mean value is not applicable. We then use One-Hot Encoding to encode the features, which is a requirement for a lot of machine learning models. This creates a binary column for each category with a value of 1 for the category the row belongs to and 0 for all other categories.

3.3 Feature Engineering

3.3.1 Temporal Features

To use temporal features we have to construct features based on the timestamp, because most machine learning models can not handle the `datetime` datatype natively. We construct the features `day_of_week`, which corresponds with numbers 0-7 for the weekdays Monday-Sunday, `day_of_year` which is between 0-365, `week_of_year` between 0-52, month between 0-12, `hour` between 0-24, and day which is the day in a given month between 0-31.

3.3.2 Building Features

We also introduce the feature `building_age` as $2017 -$ `year_built` so that we can model a relationship between the buildings age and its energy usage, which is possibly better described through the buildings age in years instead of the exact year the building was built.

4 Modeling

4.1 Multiple Linear Regression

Multiple Linear Regression is an approach to model the relationship between a dependent variable and two or more independent variables. A linear model makes a prediction by computing a weighted sum of the input features and adds a bias, which is also called intercept.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$$

In this equation, $\hat{y}$ is the predicted value, n is the number of input features, x_i is the i-th feature value, and θ_j is the j-th model parameter. Multiple linear regression has already been used to predict building energy loads. Catalina, Virgone and Blanco (2008) used regression models to predict monthly heating demand in residential buildings. Among others they used building shape factor, window-to-floor area ratio, and climate features as inputs. They found that multiple linear regression models are easy to use and efficient forecasting tools for heating demand prediction in residential buildings.

4.2 Decision Trees

As we are using the Scikit-Learn implementation in our modeling phase, we focus on the Classification and Regression Tree (CART) (Breiman, Friedman, Olshen & Stone, 2017) algorithm that the library uses to build Decision Trees. The algorithm works by first splitting the training set into two subsets using a single feature k and a threshold t_k. k and t_k are determined by minimizing the entropy in their subsets, which is the subsets purity weighted by its size. The cost function the CART algorithm minimizes for regression tasks is:

$$J(k, t_k) = \frac{m_{left}}{m}\mathrm{MSE}_{left} + \frac{m_{right}}{m}\mathrm{MSE}_{right} \text{ where } \begin{cases} \mathrm{MSE}_{node} = \sum_{i \in node}(\hat{y}_{node} - y^{(i)})^2 \\ \hat{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y^{(i)} \end{cases}$$

with m being the number of instances in the node that is to be split and MSE being the Mean Squared Error. Decision Trees can fit to the training data very closely which enables them to model complex relationships but also makes them prone to overfitting. To avoid overfitting several hyperparameters are commonly used to apply regularization. In Scikit-Learn `max_depth` (maximum depth of the decision tree), min_samples_split (minimum number of samples a node must have before it can be split), and min_samples_leaf (minimum number of samples a leaf node must have) are important hyperparameters.

4.3 Random Forest

The random forest algorithm is one of the most widespread machine learning methods (Breiman, 2001). It is an ensemble method that uses decision tree classifiers that are built with two different randomizations. The training of each individual tree is made on a subset of the training data with duplicate data points so that it has the same size as the original data set. The second randomization is attribute sampling. At each node split in the decision tree training, only a subset of the input variables is used to search for the best split. Breiman (2001) proposed $\lfloor \log_2(\#features) + 1 \rfloor$ as the number of input variables used at each split. The final prediction of the random forest model is then given by majority voting of the decision trees. One advantage of the random forest method is that the number of trees used is typically not a hyperparameter that has to be tuned because the generalization error of the random forest model converges to a limit as the number of trees is raised. One Hyperparameter that can be tuned is the maximum depth of the individual trees, which is typically done by controlling the number of features to consider when looking for the best split, the minimum number of samples required to split a node, or a hard limit for tree depth.

4.4 Gradient Boosting Methods

Boosting methods combine many weak learners into a strong learner in an iterative way. Weak learners are models that perform only slightly better than random (Freund, 2001). Gradient Boosting is a regression algorithm based on boosting (Friedman 2001). The mathematical background of Gradient Boosting is adapted from (Bentéjac, Csörgő & Martínez-Muñoz, 2021). Given a dataset $D = \{x_i, y_i\}_1^N$, the gradient boosting algorithm finds an approximation $F'(x)$ of the function $F^*(x)$, which maps inputs x to their outputs y in D. $F'(x)$ is constructed by minimizing the expected value of a loss function $L(y, F(x))$. To approximate $F^*(x)$, the gradient boosting algorithm constructs a weighted sum of functions $F_m(x) = F_{m-1} + \rho_m h_m(x)$, with ρ_m being the weight of the m-th function $h_m(x)$, which is a individual model of the ensemble, i.e. in the case of gradient boosting $h(x)$ is a decision tree. The approximation is computed by iteratively training subsequent models on the residual error of the previous model and minimizing the loss function at each iteration. This approach can suffer from overfitting if regularization is not applied adequately in the training process (Friedman 2001). The usage of some loss functions can lead to the residual error becoming zero, because the model fits the residual error of the previous model perfectly and thus stopping the algorithm prematurely. To apply regularization to the gradient boosting algorithm, shrinkage is used to reduce each gradient decent step $F_m(x) = F_{m-1}(x) + v\rho_m h_m(x)$ with $v \in (0, 1.0]$.

4.4.1 XGBoost

XGBoost (Chen & Guestrin, 2016) is a gradient boosting method based on decision trees with an explicit focus on scalablity. XGBoost also builds an additive model like Gradient boosting but only uses Decision Trees as base learners. The XGBoost algorithm minimizes the following loss function

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \quad \text{with} \quad \Omega(h) = \gamma T + \frac{\lambda \|w\|^2}{2}$$

Here T is the nomber of leaves of a tree and w are the output scores of the leaves. With γ the minimum loss reduction gain needed to split a node is controlled, which acts as a hyperparameter for regularization. XGBoost also uses randomization techniques at the tree and tree node level to reduce overfitting and reduce computing times.

4.4.2 LightGBM

LightGBM (Ke et al., 2017) is library that implements gradient boosting algorithms to be time efficient with various optimization techniques that make it suitable for large datasets. LightGBM also introduces two new features compared to previous performant implementations of gradient boosting, namely Gradient-based One-Side Sampling (GOSS), a subsampling technique that is used during creation of the training sets for the base-learners, and Exclusive Feature Bundling (EFB), a feature preprocessing technique. Both techniques improve computation times in training.

4.5 Evaluation metric

The ASHRAE Great Energy Predictor III uses the Root Mean Squared Logarithmic Error (RMSLE) as an evaluation metric. It is an adaption of the RMSE that reduces the risk that meters with a high relative consumption would influence the score compared to meters with low consumption.

$$\epsilon = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (log(p_i + 1) - log(a_i + 1))^2}$$

Where ϵ is the RMSLE value (score), n is the total number of observations in the dataset, p_i is the prediction of the traget, a_i is the actual target for i, and $log(x)$ is the natural logarithm of x.

5 Evaluation

Model	Validation	Test Set - Private	Test Set - Public	Training Time (in seconds)
Linear Regression	1.852	2.163	1.863	**56.29**
Decision Tree	0.966	1.759	1.541	122.86
Random Forest	**0.586**	1.499	1.240	326.94
XGBoost	1.408	1.664	1.240	973.82
LightGBM	0.869	**1.415**	**1.181**	89.92

Table 5.1: Results

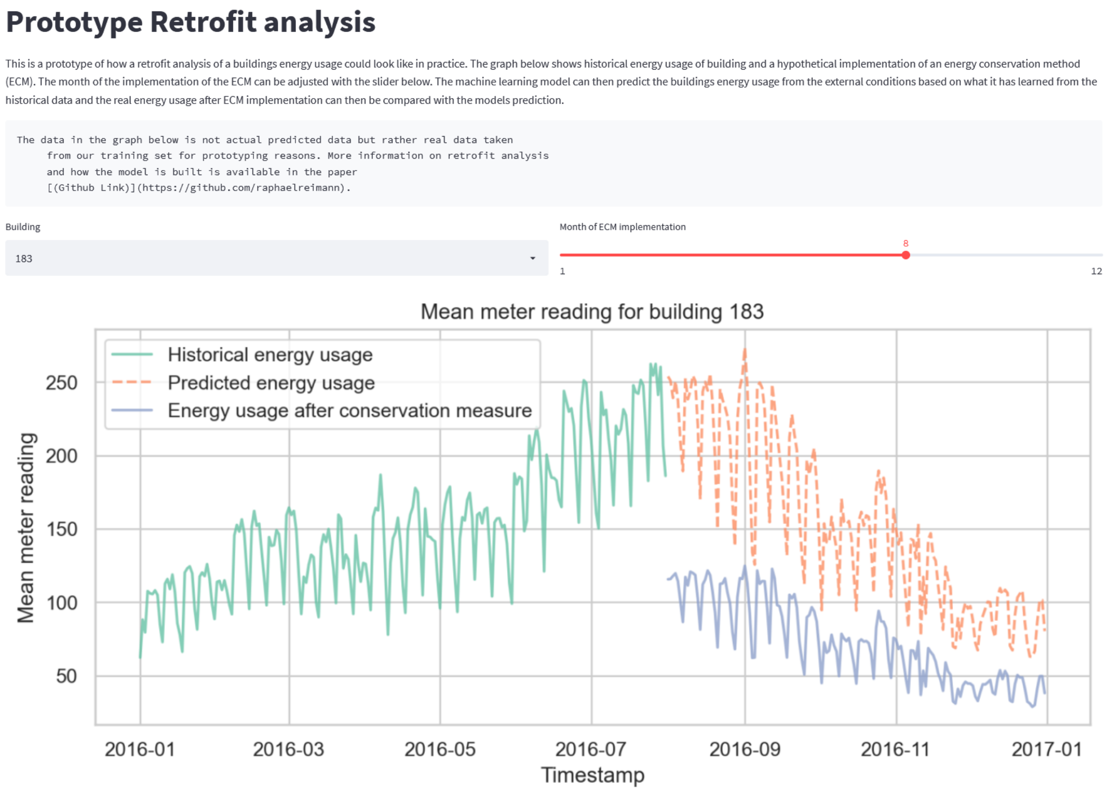
The predictions were computed with preprocessing `building_id` and `site_id` as numerical features. We also ran experiments preprocessing the id features as categorical, because the ids are categories and the relationship between building/site and assigned id is arbitrary. However, One-Hot-Encoding hundreds of different id values during preprocessing inflated the memory size of the dataframe in a way that we were not able to train all models on that data.[1]

The Validation error is the error of the prediction of the validation set we retrieve during the train-test-split. The score of the test set is retrieved as the result form the submissions from the Kaggle competition. The private test set is 49% of the test data, the public test set is 51% of the test data. We see that the Linear Regression model performs worst, which is to be expected since we are modeling complex relationships within the data that are not linear. The decision tree model performs well on the validation set and very poorly on the test set, which suggests over-fitting during training. The random forest model is the best performing model on the validation set, with an error significantly lower than the other models, which suggests overfitting. However it actually performed well on the test sets. The XGBoost model actually performs better on the public test set than on the validation set. The LightGBM model performs well on the validation set and best predicts the unseen test data. This suggests that it generalized best. The LightGBM model also trained significantly faster than the other complex model (apart from the Linear regression baseline model). A plot that compares the prediction of each model with the actual energy usage for the training set is available in the notebook `02 Modeling.ipynb`.

[1] All computations were performed on a system with an AMD Ryzen 5 3600 6-Core/12-Thread Processor with 32 GB 3600 MHz memory.

6 Deployment

To deploy our machine learning models we have built a prototype to showcase two practical use cases. The prototype is a web application that is available online at this link or can be run locally with as described in the README file. We have built both a prototype of the retrofit analysis, where we show how the prediction of the baseline energy usage of a building could look like after implementation of an energy savings measure. Note that the predicted values in the graph are not actual prediction, because the computation is slow and currently not optimize for a usable user experience. We have also implemented an energy usage calculator in the same web application where the user can change input values and see how the predicted energy usage changes.



Prototype Retrofit analysis

This is a prototype of how a retrofit analysis of a buildings energy usage could look like in practice. The graph below shows historical energy usage of building and a hypothetical implementation of an energy conservation method (ECM). The month of the implementation of the ECM can be adjusted with the slider below. The machine learning model can then predict the buildings energy usage from the external conditions based on what it has learned from the historical data and the real energy usage after ECM implementation can then be compared with the models prediction.

```
The data in the graph below is not actual predicted data but rather real data taken
     from our training set for prototyping reasons. More information on retrofit analysis
     and how the model is built is available in the paper
     [(Github Link)](https://github.com/raphaelreimann).
```

Building: 183

Month of ECM implementation: 8

Figure 6.1: Prototype Retrofit Analysis Plot

References

Amber, K. P., Aslam, M. W., Mahmood, A., Kousar, A., Younis, M. Y., Akbar, B., … Hussain, S. K. (2017, October). Energy Consumption Forecasting for University Sector Buildings. *Energies*, *10*(10), 1579. Retrieved from http://www.mdpi.com/1996-1073/10/10/1579 doi: 10.3390/en10101579

Bentéjac, C., Csörgő, A. & Martínez-Muñoz, G. (2021, March). A comparative analysis of gradient boosting algorithms. *Artif Intell Rev*, *54*(3), 1937–1967. Retrieved from https://doi.org/10.1007/s10462-020-09896-5 doi: 10.1007/s10462-020-09896-5

Borges, C., Penya, Y., Fernández, I., Prieto, J. & Bretos, O. (2013, April). Assessing Tolerance-Based Robust Short-Term Load Forecasting in Buildings. *Energies*, *6*(4), 2110–2129. Retrieved from http://www.mdpi.com/1996-1073/6/4/2110 doi: 10.3390/en6042110

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. Retrieved from http://link.springer.com/10.1023/A:1010933404324 doi: 10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (2017). *Classification And Regression Trees* (1st ed.). Routledge. Retrieved from https://www.taylorfrancis.com/books/9781351460491 doi: 10.1201/9781315139470

Burak Gunay, H., Shen, W., Newsham, G. & Ashouri, A. (2019, April). Detection and interpretation of anomalies in building energy use through inverse modeling. *Science and Technology for the Built Environment*, *25*(4), 488–503. Retrieved from https://www.tandfonline.com/doi/full/10.1080/23744731.2019.1565550 doi: 10.1080/23744731.2019.1565550

Catalina, T., Virgone, J. & Blanco, E. (2008, January). Development and validation of regression models to predict monthly heating demand for residential buildings. *Energy and Buildings*, *40*(10), 1825–1832. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0378778808000844 doi: 10.1016/j.enbuild.2008.04.001

Chen, T. & Guestrin, C. (2016, August). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). San Francisco California USA: ACM. Retrieved from https://dl.acm.org/doi/10.1145/2939672.2939785 doi: 10.1145/2939672.2939785

Cheng Fan, Fan, C., Fan, C., Yan, D., Xiao, F., Li, A., … Kang, X. (2020). Advanced data analytics for enhancing building performances: From data-driven to big data-driven approaches. *Building Simulation*, *14*(1), 3–24. doi: 10.1007/s12273-020-0723-1

Edwards, R. E., New, J. & Parker, L. E. (2012, June). Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, *49*, 591–603. Retrieved from https://linkinghub.elsevier.com/

retrieve/pii/S0378778812001582 doi: 10.1016/j.enbuild.2012.03.010

Fan, C., Xiao, F. & Wang, S. (2014, August). Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. *Applied Energy*, *127*, 1–10. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0306261914003596 doi: 10.1016/j.apenergy.2014.04.016

Freund, Y. (2001). An adaptive version of the boost by majority algorithm. *Machine Learning*, *43*(3), 293–318.

Geyer, P., Schlüter, A. & Cisar, S. (2017, January). Application of clustering for the development of retrofit strategies for large building stocks. *Advanced Engineering Informatics*, *31*, 32–47. Retrieved from https://www.sciencedirect.com/science/article/pii/S1474034616300167 doi: 10.1016/j.aei.2016.02.001

González, P. A. & Zamarreño, J. M. (2005, June). Prediction of hourly energy consumption in buildings based on a feedback artificial neural network. *Energy and Buildings*, *37*(6), 595–601. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0378778804003032 doi: 10.1016/j.enbuild.2004.09.006

Karatasou, S., Santamouris, M. & Geros, V. (2006, August). Modeling and predicting building's energy use with artificial neural networks: Methods and results. *Energy and Buildings*, *38*(8), 949–958. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0378778805002161 doi: 10.1016/j.enbuild.2005.11.005

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Marasco, D. E. & Kontokosta, C. E. (2016, September). Applications of machine learning methods to identifying and predicting building retrofit opportunities. *Energy and Buildings*, *128*, 431–441. Retrieved from https://www.sciencedirect.com/science/article/pii/S0378778816305813 doi: 10.1016/j.enbuild.2016.06.092

Marino, D. L., Amarasinghe, K. & Manic, M. (2016, October). Building energy load forecasting using Deep Neural Networks. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society* (pp. 7046–7051). Florence, Italy: IEEE. Retrieved from http://ieeexplore.ieee.org/document/7793413/ doi: 10.1109/IECON.2016.7793413

Miller, C. (2019). More Buildings Make More Generalizable Models—Benchmarking Prediction Methods on Open Electrical Meter Data. *Machine Learning and Knowledge Extraction*, *1*(3), 974–993. doi: 10.3390/make1030056

Miller, C., Arjunan, P., Kathirgamanathan, A., Fu, C., Roth, J., Park, J. Y., ... Haberl, J. (2020). The ASHRAE Great Energy Predictor III competition: Overview and results. *Science and Technology for the Built Environment*, *26*(10), 1427–1447. doi: 10.1080/23744731.2020.1795514

Miller, C., Kathirgamanathan, A., Picchetti, B., Arjunan, P., Park, J. Y., Nagy, Z., … Meggers, F. (2020). The Building Data Genome Project 2, energy meter data from the ASHRAE Great Energy Predictor III competition. *Scientific data*, *7*(1), 368. doi: 10.1038/s41597-020-00712-x

Papadopoulos, S., Azar, E., Woon, W.-L. & Kontokosta, C. E. (2018, May). Evaluation of tree-based ensemble learning algorithms for building energy performance estimation. *Journal of Building Performance Simulation*, *11*(3), 322–332. Retrieved from https://www.tandfonline.com/doi/full/10.1080/19401493.2017.1354919 doi: 10.1080/19401493.2017.1354919

Pulido-Arcas, J. A., Pérez-Fargallo, A. & Rubio-Bellido, C. (2016, December). Multivariable regression analysis to assess energy consumption and CO2 emissions in the early stages of offices design in Chile. *Energy and Buildings*, *133*, 738–753. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0378778816312671 doi: 10.1016/j.enbuild.2016.10.031

Roth, J., Chadalawada, J., Jain, R. K. & Miller, C. (2021). Uncertainty Matters: Bayesian Probabilistic Forecasting for Residential Smart Meter Prediction, Segmentation, and Behavioral Measurement and Verification. *Energies*, *14*(5), 1481. doi: 10.3390/en14051481

Tsanas, A. & Xifara, A. (2012, June). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, *49*, 560–567. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S037877881200151X doi: 10.1016/j.enbuild.2012.03.003

Tso, G. K. & Yau, K. K. (2007, September). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, *32*(9), 1761–1768. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0360544206003288 doi: 10.1016/j.energy.2006.11.010

Wang, Z., Wang, Y., Zeng, R., Srinivasan, R. S. & Ahrentzen, S. (2018, July). Random Forest based hourly building energy prediction. *Energy and Buildings*, *171*, 11–25. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0378778818311290 doi: 10.1016/j.enbuild.2018.04.008

Xuemei, L., Jin-hu, L., Lixing, D., Gang, X. & Jibin, L. (2009, July). Building Cooling Load Forecasting Model Based on LS-SVM. In *2009 Asia-Pacific Conference on Information Processing* (pp. 55–58). Shenzhen, China: IEEE. Retrieved from http://ieeexplore.ieee.org/document/5196994/ doi: 10.1109/APCIP.2009.22

Yee, T. (n.d.). *Problem with the dataset - Measurement Units.* Retrieved from https://www.kaggle.com/c/ashrae-energy-prediction/discussion/118753

Yu, Z., Haghighat, F., Fung, B. C. & Yoshino, H. (2010, October). A decision tree method for building energy demand modeling. *Energy and Buildings*, *42*(10), 1637–1646. Retrieved from https://linkinghub.elsevier.com/retrieve/

pii/S0378778810001350 doi: 10.1016/j.enbuild.2010.04.006

Zhao, H., Magoulès, F. & Magoulès, F. (2012, August). A review on the prediction of building energy consumption. *Renewable & Sustainable Energy Reviews*, *16*(6). doi: 10.1016/j.rser.2012.02.049