ASHRAE - Great Energy Predictor III

Data Science Project Protocol

Author: Avishai Mesilati

Contents

1. Introduction

This Data Science project revolves around a Kaggle competition hosted by ASHRAE (American Society of Heating and Air-Conditioning Engineers), an organization with a distinguished history dating back to 1894. ASHRAE is dedicated to advancing the realms of heating, ventilation, air conditioning, refrigeration, and their associated fields.

The drive to enhance building efficiency, minimize expenses, and reduce emissions has led to substantial investments in the construction sector. One innovative approach is pay-for-performance financing, in which building owners pay based on the disparity between their actual energy consumption and what it would have been without energy-efficient modifications. However, evaluating the value of these enhancements can be intricate because determining the exact energy consumption of an unmodified building is challenging. To address this challenge, counterfactual models are constructed. These models enable a comparison between the energy consumption of an upgraded building, which consumes less energy, and the modeled values representing the original building, thus quantifying the savings derived from retrofitting.

This project is centered on the creation of counterfactual models for chilled water energy consumption. These models are developed using historical energy usage data and observed weather conditions. The projected model values will be matched against the actual energy usage of buildings, allowing for an examination of energy and cost savings achieved through efficiency improvements.

The original dataset encompasses a year's worth of hourly meter readings from 1449 buildings situated across 16 diverse sites worldwide. These readings capture four distinct types of energy usage: electricity, chilled water, steam, and hot water.

By refining the accuracy of estimations related to energy-saving investments, we aim to inspire greater involvement from substantial investors and financial institutions. Ultimately, our efforts will contribute to advancements in building efficiencies and a more sustainable future.

This project approaches the energy prediction challenge as a regression problem. It involves forecasting the chilled water energy consumption of specific buildings on given dates within particular 8-hour time intervals.

2. Methodology (Project Design)

2.1. Data

The bulk of the dataset for this project consists of three primary types of data:

- Energy consumption readings in buildings.
- Information about the buildings themselves.
- Weather data related to the geographical areas of the buildings.

The data sourced from tables named "train", "building_metadata", and "weather_train" respectively. The weather data is collected from meteorological stations situated near the building sites, providing relevant weather information.

The dataset was obtained from “Kaggle”, a prominent platform for data science competitions, which can be accessed at the following link: ASHRAE - Great Energy Predictor III.

During the course of the Kaggle competition, geographic location data of the 16 sites where the buildings are located was leaked. Furthermore, many participants provided additional details, including university names associated with the buildings in the dataset. The reliability of this leaked data was supported by multiple winners' submissions and the consistency of their models' performance.

These 16 sites are distributed across the USA, England, Ireland, and Canada, with the majority situated in the USA. The geographical information enabled the generation of supplemental relevant data, such as latitude and longitude for each site. This additional data facilitated the derivation of the area's elevation, the beginning and end of "daylight saving time" (DST), as well as seasonal and holiday periods.

The enriched dataset will be pivotal in our efforts to develop robust counterfactual models for chilled water energy usage and assess energy and cost savings effectively.

2.1.1. Data processing and flat file creation

Data processing and flat file creation are fundamental steps in this project, conducted using a combination of SQL and R. I initiated the process by extracting information from the original tables and enriching the data. This resulted in the creation of three new tables: "train_p", "building_metadata_p" and "weather_train_p". Notably, the "train_p" and "weather_train_p" tables, which contained timestamp variables, underwent an additional transformation in an R notebook, aggregating data into eight-hour time intervals. Furthermore, by combining these two tables with "building_metadata", I generated a final flat file named "ff_train_agg".

Key Issues and Challenges in the process

In the SQL portion of the project, several key issues and challenges were addressed:

Creation of "site_ids_train" Table: To enhance the dataset, I began by creating this table, which served as the foundation for various auxiliary variables, including:

- "utc_st_offs": An offset based on site location that standardizes universal time (UTC) to local time. This was crucial for unifying timestamp variables in the "train" and "weather_train" tables, with "train" representing local time and "weather_train" representing UTC.
- "start_dst_16" and "end_dst_16": Variables denoting the start and end of "daylight saving time" (DST) at each site, enabling accurate conversion of universal time to local time.

Another variable created here is indicated in the Data Enrichment section.

1. **Filtering of "train" Table**: The "train" table contained 20.2 million records, representing four types of energy readings. For my project's focus on chilled water energy consumption, I filtered the table to retain only "chilled water" records, reducing the dataset to approximately 4.2 million records. This process also involved omitting data from six sites that did not utilize chilled water energy. After excluding those 6 sites, the dataset comprises the remaining 10 sites, with 9 located in the United States and 1 in Canada.
2. **Handling "0" Energy Readings**: In the "train_p" table, we excluded energy readings with a value of "0," leaving around 3.5 million records. These "0" values were identified as incorrect data based on information from external sources.
3. **One-Hot Encoding for "building_metadata_p" Table**: The "primary_use" column in the "building_metadata_p" table underwent "One-Hot Encoding." Two categories were removed from this column as they yielded a single unique value during the encoding process.
4. **Data Processing for "weather_train_p" Table:** On this table, I made a join with the "site_ids_train" table, and created several new variables. One of these variables was a temporary variable for the local timestamp, based on the universal timestamp. This conversion was necessary to allow for future table joins. Three more variables created during this process are discussed in in the Data Enrichment section.
   I also conducted specific data processing within the "weather_train_p" table to enhance data quality. In the "cloud_coverage" variable, I converted instances of the value "9" to 'NULL'. This transformation was based on insights from Kaggle forum discussions and other online sources, which indicated that "9" values represent missing or unavailable cloud coverage data for the current observation. Similarly, in the "precip_depth_1_hr" variable, I transformed occurrences of "-1" to 'NULL' based on the same reasoning, signifying that precipitation data was either missing or not available for the current observation.

5. **Generation of New Variables**: While creating the "ff_train_agg" flat file through SQL joins, 11 new variables were generated. Addressing these variables is provided in the Data Enrichment section.

2.1.2. Inclusion criteria

- The dataset contains chilled water type energy measurements.

- The timestamp variable is in local time in the “train” table and in UTC in the “weather_train” table.
- All data is available.
- The data pertains to the countries where the research was conducted.
- [1]The data corresponds to the year 2016 only.

2.1.3. Exclusion criteria

- Cases that occurred before January 1, 2016, or after December 31, 2016.

2.1.4. Outcome (y)

The outcome variable is referred to as "meter_reading_sum" and is derived by aggregating the eight-hour readings using the 'sum' operation. This variable represents the energy consumption of chilled water for a

particular building on a specific date within an 8-hour time interval.

2.1.5. “EDA” strategy

Exploratory Data Analysis (EDA) was conducted on the eight-hour time interval flat file using a combination of statistical analysis and visual examination in R. The initial phase involved utilizing functions such as "summary," "Table1," and "exploreData" to gain preliminary insights into the data.

Correlation tests were applied to assess the relationships between variables. Numerical variables, which encompassed ordinal and binary types, were subjected to "corr.test," while nominal variables, including binary variables, were analyzed using "cv.test." In each test output, pairs of variables exhibiting an absolute correlation coefficient of 0.7 or greater and statistically significant results were identified. To avoid multicollinearity, one variable from each highly correlated pair is a candidate for later removal.

Furthermore, pairs of variables with absolute correlation coefficients falling between 0.2 and 0.7 and displaying statistical significance were also retained. This strategic filtering was executed to prepare for a subsequent analysis aimed at identifying distinctive patterns within graphs that portray the relationships between variables. The primary goal of this analysis was to discover possibilities for generating innovative and informative variables from these graphs.

In addition to statistical assessments, data visualization was employed to complement the quantitative findings. Scatter plots, box plots, and heat maps were generated to provide visual representations that facilitate a deeper understanding of the data and potential insights beyond statistical analysis.

[1] The code in its current state will not run on a period other than 2016, but it could easily be generalized to run on any year.

2.1.6. Outlier Detection and Treatment

In order to determine which outliers to retain and which to remove from the dataset, I conducted comprehensive outlier detection and treatment using R. This process involved two key steps:

Kolmogorov-Smirnov Test: I employed the Kolmogorov-Smirnov statistical test to identify potential outliers in the data.

Correlation Impact Test: Additionally, I assessed the influence of outliers on the correlation between the tested variable and the outcome variable using the "cocor" function.

Based on the findings of these tests, decisions were made regarding which outliers would be retained and which would be removed. In cases where outliers were deemed relevant and should be retained, they were further categorized or underwent transformation, as required. Categorization was applied to variables with a limited value range, while variables with a broad value range, typically exhibiting a tail in their distribution, were transformed using logarithmic or square root operations.

2.1.7. Missing data

In the initial stages, before performing data aggregation on two processed original tables, a check for missing values was conducted. The purpose of addressing missing data at this stage, was to ensure that aggregation would not carried out while excluding the missing rows, as if they were dropped from the dataset. In the later stages of the “EDA” process, a thorough examination of missing values was performed on the entire dataset. Following this, in the “Data Cleansing” stage, an additional round of missing data management was conducted after addressing outliers, which also introduced new missing values. The following steps were taken:

Columns with over 79% of missing values were identified and subsequently removed.

Rows that included columns with less than 1% of missing values were also eliminated, and also rows that had more than 50% of missingness in the row itself.

The type of mechanism responsible for the formation of missing values was determined. Variables categorized as "MNAR" underwent a categorization process, while variables classified as "MAR" and "MCAR" were intended to the "KNN" imputation method.

2.1.8 Time Series Handling

I examined all the variables in the dataset to determine their stationarity. Subsequently, I intended to utilize the "Vector Auto Regression" (VAR) function in R to address non-stationary variables. The objective was to identify the significant lags for each variable and transform the dataset into a conventional tabular format. This transformation involved carrying data forward from the significant lags to the current row and storing it in new columns. However, I found that all non-stationary variables lacked autocorrelation, rendering the application of the 'VAR' model unfeasible, and as a result, the dataset remained unaltered.

2.1.9. Data enrichment

Throughout various stages of the project, I enriched the data by generating additional features:

In the process of creating the flat file:

In the "site_ids_train" table, I calculated the surface elevation for each of the 16 sites. Calculating the elevations was made using an internet application based on longitude and latitude that I found for each geographical point. Longitude and latitude were taken from Google Maps based on university locations. When university information was not available, I randomly selected points within the city and and calculated elevation based on their geographical coordinates.

In the "building_metadata_p" table, I applied "One-Hot Encoding" to the "primary_use" feature.

In the "weather_train_p" table, I introduced features for date and time ranges, dividing the day into three 8-hour partitions. Additionally, I created a relative humidity feature.

While creating the "ff_train_agg" flat file through a join process, I derived several features from the date, including month, day of the year, day of the week, and indicators for working hours and weekends. I also incorporated a feature for seasons, based on a review of the season dates for each city, and an indicator for holidays, determined from holiday dates within each country.

Cyclic Features Transformation:

Some of these features that are cyclic were replaced by features based on them, that represent the horizontal and vertical components of their polar representation, using cosine and sine functions. This was done to ensure that the models interpret these variables as cyclic, preventing erroneous interpretations where, for example, the model interprets the distance between wind directions 350 and 360 as shorter than the distance between 10 and 360, which is not the case in practice.

Aggregations on 8-Hour Intervals:

Data enrichment involved applying various aggregation methods, such as sum, minimum, maximum, standard deviation, mean, and mode, to create new features at 8-hour intervals. Different features underwent specific aggregation types, contributing to the expansion of the dataset.

In Data Cleansing Process:

I created binary features based on patterns observed in graphs representing relationships between combinations of variables in the data.

Binary zero indicator features were generated in response to the presence of sparse columns.

I generated a binary feature designed to assist the model in distinguishing between even and odd values within two cloud_coverage features due to their unique behavior.

Some features were replaced with square root transformations to ensure a normal distribution, which was essential for outlier handling.

Following the categorization of "MNAR" (Missing Not at Random) variables into quartiles, I generated binary features to distinguish between zeros attributed to missing values and other non-zero values. This differentiation aids the model in recognizing values resulting from categorization versus non-process-related zeros.

[2]**Feature**	**Description**
elevation	Surface elevation
primary_use	One-Hot Encoding
date	Date
time_range	Dividing the day into three 8-hour blocks
rel_humid	Relative Humidity
month	Month
day_of_year	Day of the year
day_of_week	Day of the week
is_working_hours	Indicator for working hours
is_weekend	Indicator for a weekend
season	Season
is_holiday	Indicator for a holiday
is_upward_date	Indicator for a pattern in a graph
is_precip_dew_temperature	Indicator for a pattern in a graph
is_precip_air_temperature	Indicator for a pattern in a graph
is_zero_precip_depth	Indicator for a zero in a sparse column
dew_temperature_sqrt	Square root values
sea_level_pressure_sqr	Square root values
rel_humid_sqrt	Square root values
is_missing_year_built	Distinguishes between zeros and categorization results values
is_missing_sea_level_pressure	Distinguishes between zeros and categorization results values
is_missing_dew_temperature_sqrt	Distinguishes between zeros and categorization results values
is_missing_sea_level_pressure_sqrt	Distinguishes between zeros and categorization results values
is_missing_rel_humid_sqrt	Distinguishes between zeros and categorization results values

Table 1 – Data Enrichment

[2] The table does not include a breakdown of the types of aggregation performed.

2.1.10. Feature Selection

Feature selection was conducted in Python through a two-stage process, combining correlation analysis and model-based selection:

Correlations-based: First, features were classified into two groups - continuous/ordinal variables and nominal variables. Correlations were assessed using different methods for each group: Spearman's rank correlation for the first group and Cramer's V for the second group. When high correlations were detected between pairs of variables, one of them was dropped. In the first group, the drop decision was based on the variable's correlation with the target variable, favoring those with lower correlation. In the second group, the selection was made randomly since Cramer's V cannot be applied to continuous target variables. The identified features were promptly removed from the dataset.

Models-based: Initially, I explored the "SelectFromModel" approach using various models, which provide feature importance technique. However, one model encountered error and couldn't run, while others selected a very limited number of features from the extensive dataset. The results were highly sensitive to parameter values, leading to substantial variations in outcomes.

Subsequently, I adopted a different strategy for model-based feature selection. After the model training phase and the selection of the best-performing model, hyperparameter fine-tuning was carried out. Using the optimized parameter set, feature selection was applied to the remaining dataset. The variables that were not chosen were then dropped, resulting in a refined dataset with the selected features, ready for further analysis and modeling.

2.2. Models

2.2.1. Dataset subsetting, splitting and further processing

After conducting a qualitative PCA analysis, which is visually presented below (based on a 10% sample of data from 10 randomly selected buildings, where a PCA model was applied to their data records), a decision was made to train separate models for each building.

In the 'PCA' visualization, it's evident that certain buildings exhibit a spatial separation in their representations, indicating significant differences in their distributions. Even though some overlap existed in the representations of certain buildings, training a single model for all buildings could potentially lead to underfitting due to the complexity and distribution differences. Hence, the approach of training distinct models for each building was chosen to optimize performance and predictive accuracy by capturing the unique characteristics and variations specific to each building's data.

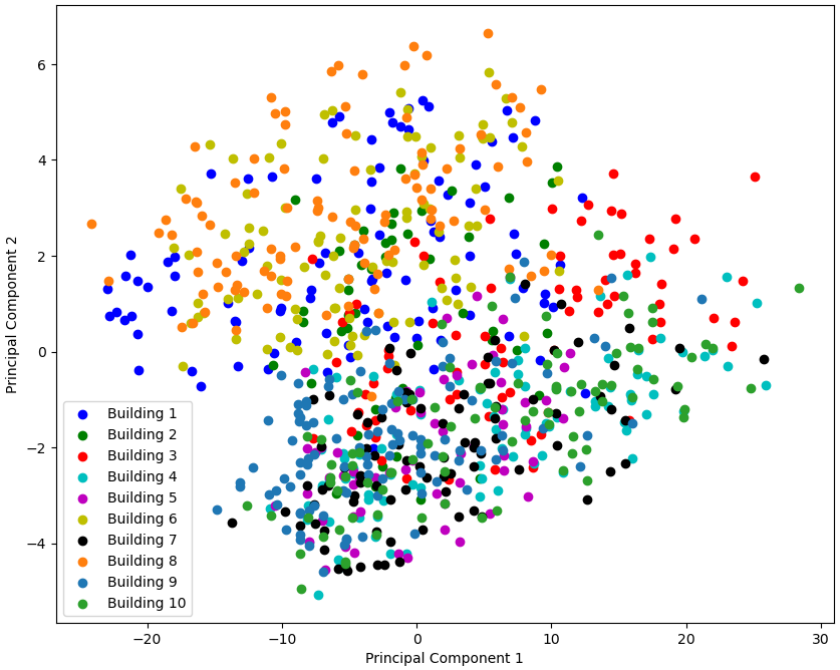

Figure 1 – PCA Visualization

In this framework, each building's data subset was further divided into train, validation, and test sets. Given the time dimension in the data, a decision was made to perform chronological splits for each building, allocating 70% of the data to the train set, 15% to the validation set, and 15% to the test set (with the train split preceding the validation and test splits). This chronological splitting preserved the data's temporal order, closely mimicking the real-world scenario of making predictions based on historical data, allowing for evaluations aligned with the intended use of the models. Additionally, chronological splitting served to prevent data leakage, as the models never had access to future data during training, ensuring robust evaluations. Note that the chronological splitting constraint made it infeasible to balance the target variable distribution among the splits. This is because resampling or mixing the splits to achieve balance would introduce data leakage.

Because of the scales of the features and the target variable, I encountered issues with the convergence of the training algorithms, due to computational difficulties. To overcome it, I implemented scaling for both the features and the target variable (I scalled all data splits according to the mean and variance based on the training set to avoid data leakage), so the problem resolved.

2.2.2. Model selection

Consistent with the strategy of training distinct models for each building, following the data splitting, I trained multiple model types on my training data using their default hyperparameters.

These models included linear regression, decision tree regressor, random forest regressor, SVR with multiple kernels, ADABOOST regressor, and gradient boosting regressor. To prioritize these models, I selected a single evaluation metric, opting for Mean Squared Error (MSE) due to the absence of outliers and the nature of the task (regression). The model exhibiting the lowest MSE on the validation split was chosen for subsequent hyperparameter tuning.

After selecting the optimal model, I employed grid search to fine-tune its hyperparameters. The sets of parameters were prioritized based on the MSE results from the validation split.

Subsequently, for each model, I assessed feature importance. In the case of LR-based models, I considered the regression coefficients, while for tree-based models, I relied on impurity-based feature importance. I then proceeded to select all features with importance scores exceeding the median importance score, essentially retaining the more influential half of the feature set. The model was refitted using this reduced feature set.

2.3. Deployment of model

The quality assurance process for this project will be conducted by a team of data scientists and civil engineers. Their responsibilities will encompass reviewing every phase of the project to guarantee the validity and reliability of the data, models, and predictions. The aspects that will be assessed include the quality of the data before the proposed processing, taking into account factors like measurement quality and continuity. Additionally, the model types will be assessed, considering that different models with varying expressive capabilities may yield diverse results. The evaluation metric will also be scrutinized, as it depends on the scale of the data and the presence of outliers, necessitating a distinct approach to evaluation.

The QA protocol for each step of the project includes the following key aspects:

Data Collection:

- Verify the reliability and consistency of data sources.
- Ensure data completeness, cleanliness, and correct formatting.

Data Splitting:

- Confirm that the train, validation, and test sets are chronologically ordered to prevent data leakage.
- Assess the adequacy of data volume for proper model fitting and reliable testing (recognizing that conclusive model fitness evaluation occurs after training).

Model Training:

- Validate that models are trained on the train set and evaluated using the validation set.
- Monitor the training process to prevent overfitting or underfitting.
- Save the best model for each building based on validation scores.

Model Testing:

- Confirm that models are tested on the test set and report the test scores for each building.
- Analyze errors, residuals, and identify any outliers or anomalies.

Model Deployment:

- Ensure models are deployed on a secure and scalable platform capable of handling prediction requests from end-users.

The end users of the predictions are construction companies striving to showcase the additional value of their buildings to potential customers. They aim to persuade customers of the benefits of their structures, particularly the significant energy savings they offer in comparison to conventional buildings lacking green energy systems. These companies provide historical data encompassing various building attributes and environmental factors. Using the customized models tailored to their specific projects, they can predict electricity usage in these buildings. This allows them to make a compelling case to customers about the advantages of their energy-efficient buildings, thus justifying higher prices and providing a valuable selling point.

The predictions will be conveyed to the end user through a user-friendly dashboard or detailed reports. These presentations will showcase the forecasted energy consumption for their specific building over a defined time frame, based on the input features they provide. To offer a better understanding of the model's performance, the predictions will be based on a test set that the model has never been exposed to, resembling the test set in our framework. This allows the user to gauge the expected prediction error for future data and new, previously unseen data.

Furthermore, the dashboard or report will provide insights into the factors influencing the predictions and their corresponding values. This approach enhances the model's explainability, offering a more detailed rationale for its predictions and empowering the end user with valuable information.

The end user will undergo a training process facilitated by a knowledgeable data scientist or engineer. During this training, they will gain insights into how the model functions, the underlying assumptions and limitations, and how to effectively interpret the dashboard or report. The training will encompass the following aspects:

Model Understanding: The user will learn how the model operates, its core principles, and how it makes predictions based on the provided input features.

Assumptions and Limitations: Understanding the model's assumptions and limitations is crucial. The user will be informed about the boundaries within which the model is reliable.

Dashboard and Report Interpretation: The user will be guided on how to read and comprehend the information presented in the dashboard or report. They will learn how to extract valuable insights from the predictions and related features.

Decision-Making and Actions: The training will cover how to utilize the predictions to make informed decisions and take appropriate actions to optimize energy consumption.

Feedback and Suggestions: Users will be encouraged to provide feedback and suggestions for model improvement. Their insights can help refine the system over time.

It's important to note that a unique model is created for each user (building), tailored to their specific requirements. However, the selection of model types and the underlying framework will be periodically updated to incorporate new models and architectures as they become available. This ensures that users continue to benefit from the latest advancements in the field.

3. Results

3.1.1. EDA and data cleansing

During the initial missingness check, before aggregating the two processed original tables, missing values were identified in nine variables. These variables fell into two distinct categories for handling:

Category 1 - Variables with gradual value changes, necessitating imputation via linear interpolation based on time sequence. There were seven variables in this category, and they were promptly imputed.

Category 2 - Variables with non-gradual value changes. These variables were handled by reducing missing values to zero through the eight-hour aggregation and by deleting corresponding rows during the data cleansing stage. This category included two variables.

In the first EDA phase, a correlation test on continuous and ordinal features revealed 29 pairs of features with an absolute correlation greater than 0.7. Furthermore, the "Cramer's V" test on nominal features found 17 pairs within the same correlation range.

A total of 33 variables were identified with missing values in this phase. Two of these variables had missing values exceeding 65%, while the dataset contained approximately 2.3% complete rows.

During the 'Data Cleansing' stage, 16 variables with outliers were designated to remain in the dataset, while 16 were dropped and replaced with missing values. Among those that remained, 11 underwent categorization into quartiles, one was transformed logarithmically, and four were retained in their original form.

After addressing outliers, which introduced additional missing values, 40 variables were found to have missing data. Out of these, 16,242 records were dropped because they included columns with 1% or less missing values. No rows were found to have over 50% missing values, and therefore, there was no need to remove rows on this basis. "floor_count" exhibited a high missingness of 96.5% and was consequently removed from the dataset.

The 12 variables with missing values that remained were all classified as "MNAR" type. These variables underwent categorization. However, five columns were handled differently. The four "MNAR" "wind_speed" columns in the dataset had up to 3.3% missing values in each. To maintain their continuous nature, they were not categorized. Instead, rows with missing values in these columns were deleted. Another column, "cloud_coverage_min," was also treated differently. Due to a high correlation (83%) with another "cloud_coverage" feature, it was dropped from the dataset. The remaining seven "MNAR" columns were categorized by quartiles. Following the completion of the categorization process, the dataset no longer contained missing values, obviating the need for further imputation methods.

3.1.2. Data enrichment

As for the various data enrichment techniques described in the methodology paragraph, I will note that during the process of creating the flat file, I introduced 32 new features to enhance the dataset. Of these, 14 were generated through one-hot encoding applied to the "primary_use" feature, and the remaining 12 resulted from the creation of 6 pairs of horizontal and vertical components derived from cyclic features' polar representations.

Furthermore, within the flat file creation process, I conducted aggregations that yielded 36 additional features. In this process, various types of aggregations were executed, enhancing the dataset's diversity.

As I transitioned into the data cleansing stage, 18 more features were incorporated. The majority of these features serve as indicators of missing values, which were produced as a result of the categorization process during missing data management.

I will note that some of these new features introduced during the data enrichment process replaced existing features in the dataset.

3.1.3. Feature selection, Regressions and Statistics

As outlined in section 2.1.10, feature selection consisted of a two-step process. The initial stage focused on correlation-based selection, and the second stage involved model-based selection conducted at the modeling phase. In the initial stage, 24 variables were removed. The list of these eliminated variables is provided below:

Eliminated Variables in Correlation-Based Feature Selection:

is_working_hours	day_of_year_y
air_temperature_mean	is_upward_date
air_temperature_max	season_x
dew_temperature_min	is_zero_precip_depth_sum
dew_temperature_mean	is_zero_precip_depth_max
sea_level_pressure_min	is_zero_precip_depth_sd
sea_level_pressure_mean	is_missing_sea_level_pressure_mean
wind_speed_min	is_missing_sea_level_pressure_max
wind_speed_max	rel_humid_mean
cloud_coverage_max	year_built
day_of_week_x	precip_depth_sum
month_x	is_zero_precip_depth_min

Table 2 – Eliminated Features – correlation-based

Regarding model-based feature selection, the graph below illustrates the proportion of the most dominant variables among the various models fitted to each building, during the model-based feature selection process. This result leads to a significant conclusion: many of the most

commonly selected important features, chosen by almost all buildings, are weather-related. Notable features include precip_depth_sd, wind_direction_y_mode, air_temperature_min, time_range_y_mode, air_temperature_y_mode, and month_y.

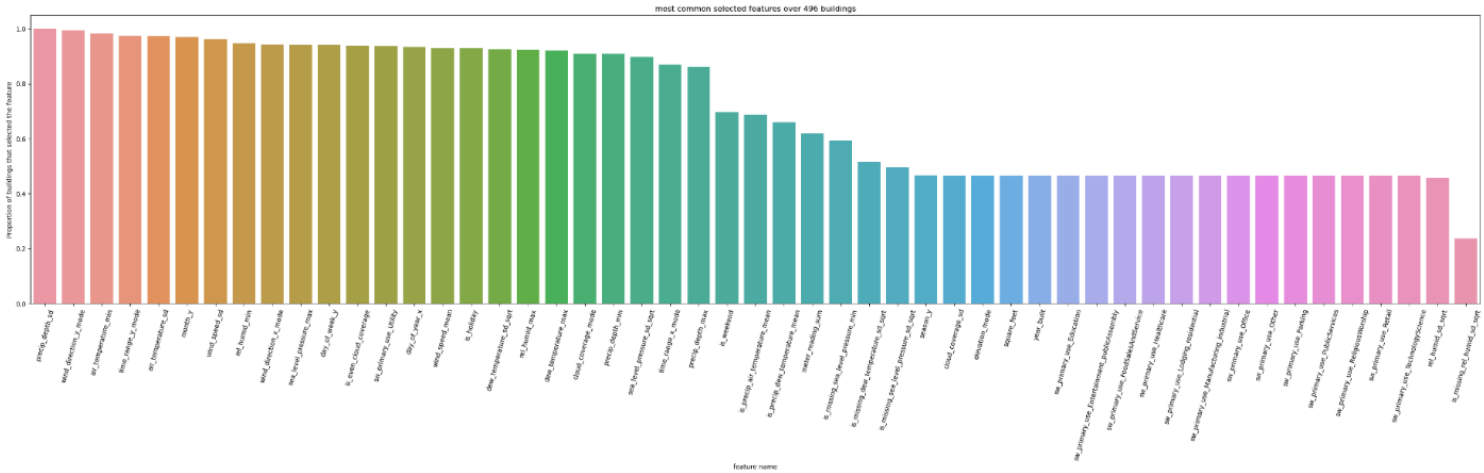

Figure 2 – Popularity of Features Among the Buildings

In the boxplot below, we can observe the "MSE" results for all three splits across all 496 buildings. Several conclusions can be drawn from this plot. First, there is no significant difference between the test "MSE" and the train "MSE," indicating that severe overfitting is not a concern in the models. Second, the validation results closely resemble the test results, providing confidence that the model's performance on the validation set is a reliable indicator of its expected real-world performance. This reliability is crucial for informed decisions regarding model deployment or its intended use. Furthermore, the similar interval sizes between the different splits suggest that these results can be considered a strong indicator for the performance of future building models.

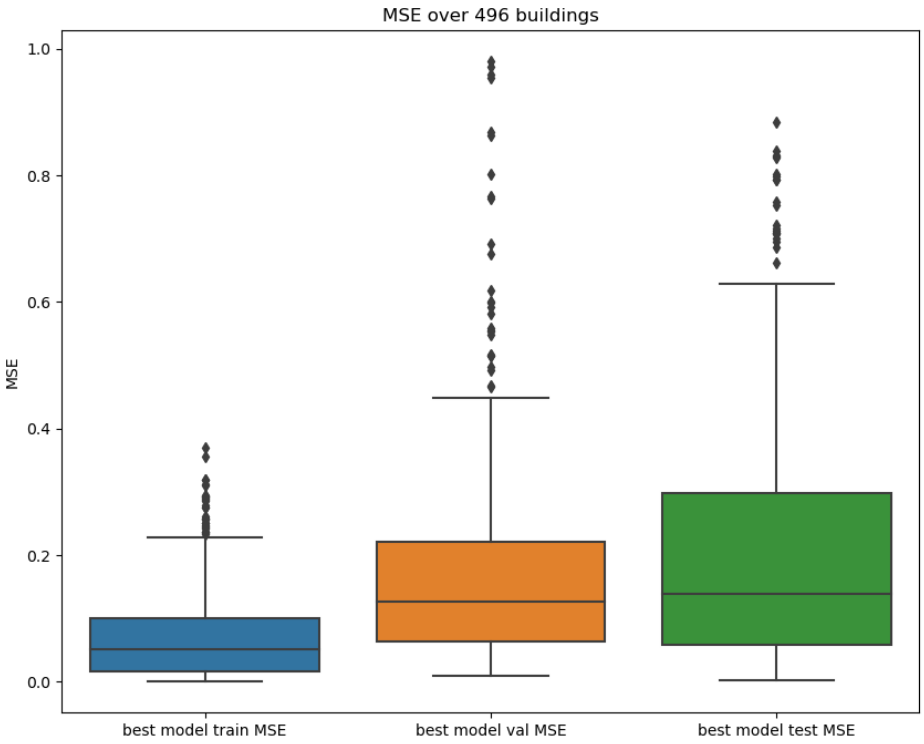

Figure 3 – MSE Distribution in train, val and test

3.1.4. Data subsetting

Each building comprises approximately 1090 records, with the data divided into 70% for training, 15% for validation, and 15% for the test set. The test set was extracted first. As mentioned previously, chronological splitting precludes the resampling of different splits to balance the distribution of the target variable across them, as this would introduce data leakage. The rationale for employing chronological data splitting is elaborated upon in section 2.2.1.

3.1.5. Model selection

The framework I developed facilitated individual model selection for each building. Below, the frequency of each model type being chosen as the best model for a building is presented.

It is evident that the “Gradient Boosting Regressor” and the “SVR” are the most frequently selected. In contrast, the “Decision Tree Regressor” and the “ADABOOST Regressor” are the least preferred choices. This diversity in model selection among different buildings underscores the importance of training customized models for each building.

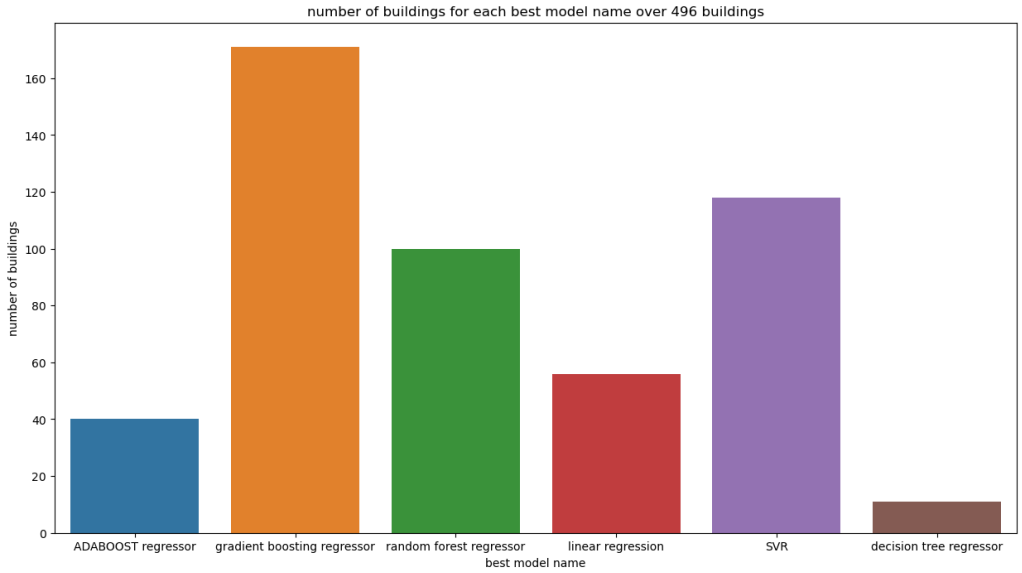

Figure 4 – Popularity of the Best Models Among the Buildings

4. Discussion and Conclusion

In this project, the primary objective was to develop a model capable of predicting chilled water energy consumption in specific buildings on given dates within specific 8-hour time intervals. This predictive task was approached as a regression problem, with the original data initially provided at an hourly level. To streamline the prediction task, all time-related data was aggregated to 8-hour intervals during the flat file creation process.

The project commenced with an extensive information-gathering phase, involving thorough exploration of data-related resources. Initially, a significant challenge was the lack of data pertaining to the exact locations of the monitored sites. However, the acquisition of data shared among Kaggle competitors played a crucial role in enriching the dataset. With this newfound information, I proceeded to incorporate various features related to the site locations, such as longitude, latitude, surface elevation, seasonal information, and holiday dates specific to each country. Additionally, I discovered the start and end dates of daylight saving time (DST) in different countries during my exploration to enrich the data. This information was crucial for converting universal time (UTC) to local time, ensuring accurate data integration.

Throughout the project, extensive exploratory data analysis (EDA) was conducted, which involved a careful handling of outliers and missing values. Special attention was given to unique cases within these outliers and missing values. During the data cleansing stage, all missing values were meticulously handled, leaving no gaps unattended. Thanks to the required actions taken during the process, there were no remaining missing values, obviating the need for K-nearest neighbors (KNN) imputations.

Regarding the time series issue, after conducting several hypothesis tests, it was discovered that some variables were non-stationary. However, it's worth noting that in these variables, there was no autocorrelation present. Consequently, I did not employ the vector auto regression (VAR) function to identify significant lags and generate new variables based on them. It's possible that the aggregations I performed at 8-hour intervals reset the autocorrelations in the variables.

As mentioned earlier, the dataset contains data from various buildings, which led to the question of whether to create a universal model for all buildings or tailor models for each one. A PCA-based analysis, which visually depicted the data distribution among different buildings, revealed significant distinctions among some of them. Consequently, I opted to develop individualized models for each building to ensure their independence.

For each building, its data (which is approximately 1090 records) was divided into three segments: training (70%), validation (15%), and test (15%). The best model is selected from a list of models based on different principles, such as linear regression and decision trees.Training was performed on the training set, while model adjustments, including hyperparameter tuning and feature selection, took place on the validation set.

The feature selection process involved identifying the most relevant features, with the selection criteria varying according to the model type (coefficient values for linear regression and the feature importance technique for the other models). After selecting the most significant features, the model was retrained on this refined feature subset. The performance of the final

model was assessed using Mean Squared Error (MSE) on the test set, along with evaluations on the training and validation sets to gain additional insights.

Limitations (of results):

The data included information from the USA and Canada exclusively.

The data covered the year 2016 only.

Predictions were provided at 8-hour intervals.

The models employed in this study were: "linear regression," "decision tree regressor," "random forest regressor," "SVR" (with multiple kernels), "ADABOOST regressor," and "gradient boosting regressor." The utilization of more advanced and expressive models might have potentially yielded better results.

A temporal discontinuity was introduced due to the removal of rows during the handling of missing values.

Despite the outlined limitations, this project has delivered a comprehensive pipeline for the development of a predictive model. This model offers valuable insights into chilled water energy consumption within specific buildings by utilizing a diverse set of features and machine learning algorithms.

Appendix: Data Retrieval Protocol

https://vanl.ink/ZGtu0