BUILDING ENERGY EFFICIENCY ANALYSIS VIA REGRESSION AND FEATURE SELECTION

CPS 595 : Software project -Final Report

Anjana Gunathilake
gunathilakea1@udayton.edu

Building Energy Efficiency Analysis via Regression and Feature Selection

Introduction

Energy Use Intensity (EUI) and ENERGY STAR ratings are 2 of most common ways to identify energy efficiency of a building. Energy use intensity (EUI) is an indicator of the energy efficiency of a building's design and/or operations. EUI states as energy per square foot per year. It is calculated by dividing the total energy consumed by the building in one year (measured in kBtu or GJ) by the total gross floor area of the building (measured in square feet or square meters).

Usually, a low EUI implies good energy performance of a building. It is important to remember that EUI varies with building type. A hospital or laboratory will have a higher EUI than an elementary school.

On the other hand, Energy Star, rates building energy performance by normalizing annual energy use, as well as building type, size, location, and other operational and general asset characteristics. It is a score between 1-100. It is showing the building's energy consumption measures up against similar buildings nationwide. Usually, a high Energy Star rating represents high energy performance building.

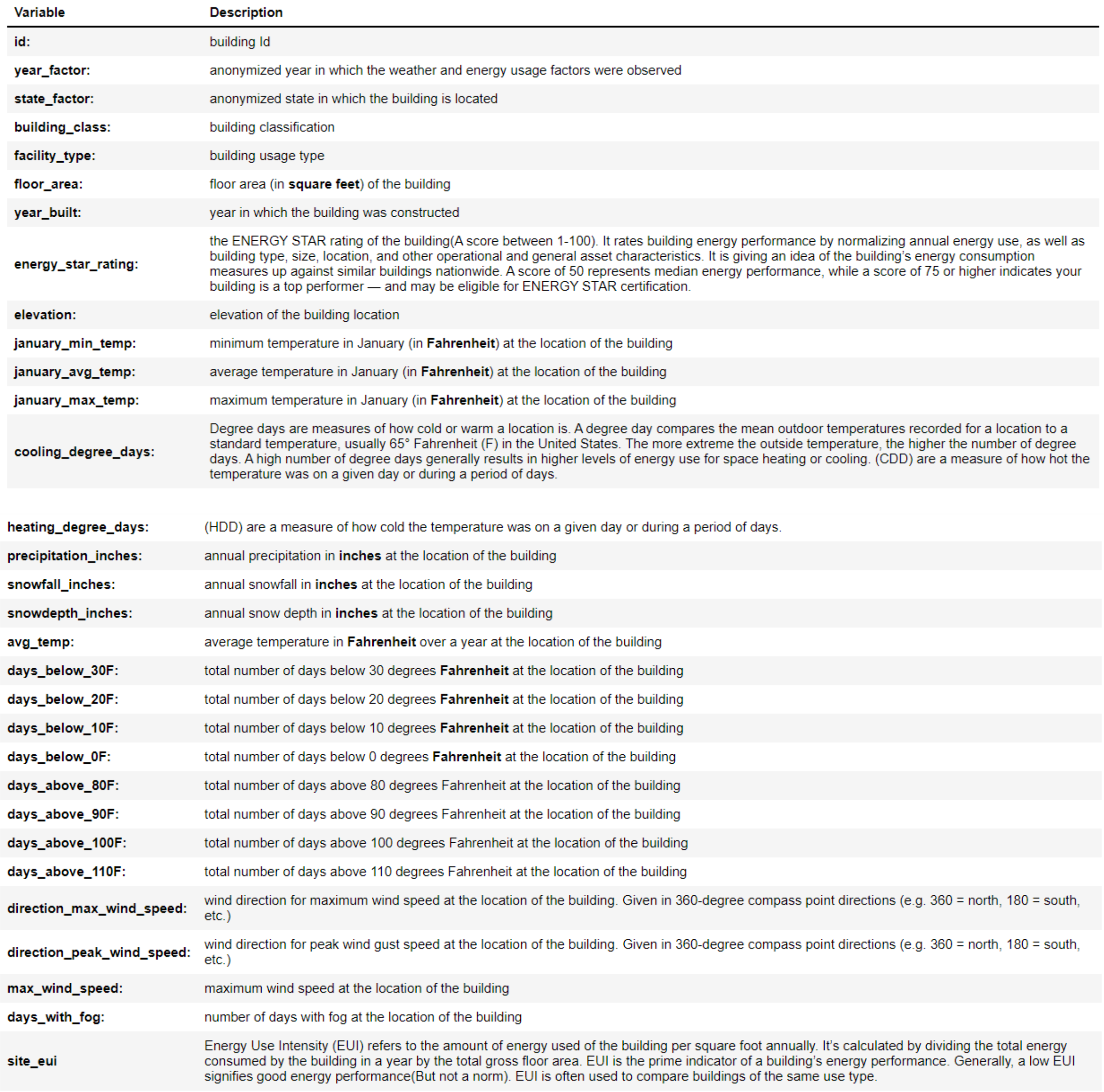
By looking at the simple interpretation of these 2 measures we can say that site EUI has a better correlation between floor area of the building and Energy Star rating, if it is available for that building. The dataset which I used for this analysis has more features other than floor area and Energy Star rating of the building.

The main goal of this project is to predict the site EUI for unseen data by using machine learning algorithms. Machine learning has two types of predictive models. They are Classification models, which predict class membership, and Regression models which predict a number. Since my task is predicting numbers, I use regression models for this project. There are various types of regression models available in machine learning. I used few of the most widely used predictive models for this project. They are Linear Regression, Decision Tree Regression, Random Forest Regression, eXtreme Gradient Boosting Regression and Categorical Boosting Regression. I tried a few other machine learning models such as Support Vector Regression and Gradient Boosting Regression. But I could not complete a complete analysis within the available time.

The next goal of this project is to analyze the performances of each model by going through the model improvement techniques. There are various model improvement techniques are available in Machine Learning. But I only went through Hyperparameter Tuning, Cross Validation and Feature Selection.

Data Description

The dataset used for the analysis was collected from Kaggle WiDS Datathon 2022 Competition, and it was created in collaboration with Climate Change AI (CCAI) and Lawrence Berkeley National Laboratory (Berkeley Lab). The dataset includes roughly 76, 000 observations of building energy usage records, building characteristics and site climate and weather data collected over 7 years and several states within the United States.

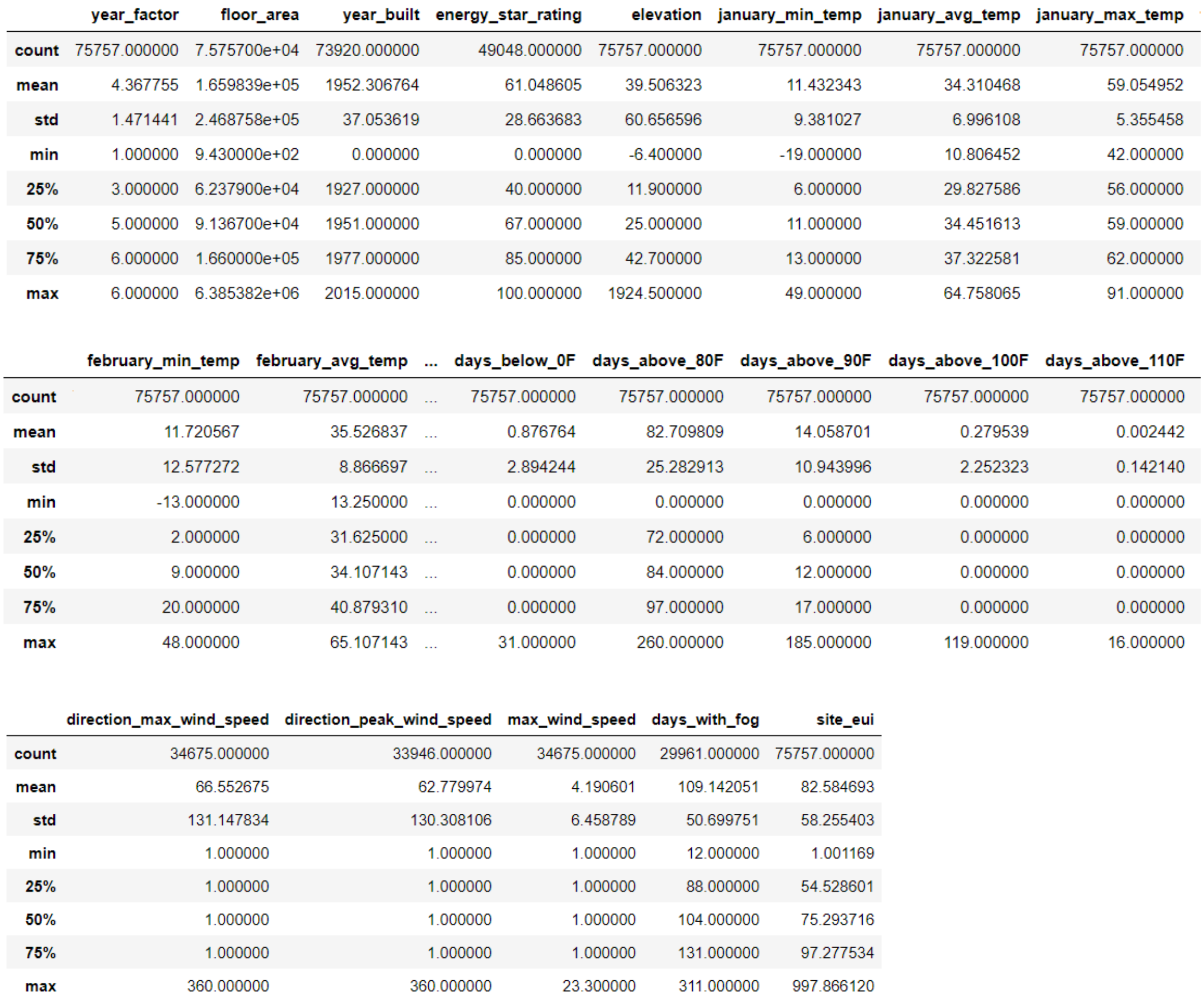


Variable	Description
id:	building Id
year_factor:	anonymized year in which the weather and energy usage factors were observed
state_factor:	anonymized state in which the building is located
building_class:	building classification
facility_type:	building usage type
floor_area:	floor area (in **square feet**) of the building
year_built:	year in which the building was constructed
energy_star_rating:	the ENERGY STAR rating of the building(A score between 1-100). It rates building energy performance by normalizing annual energy use, as well as building type, size, location, and other operational and general asset characteristics. It is giving an idea of the building's energy consumption measures up against similar buildings nationwide. A score of 50 represents median energy performance, while a score of 75 or higher indicates your building is a top performer — and may be eligible for ENERGY STAR certification.
elevation:	elevation of the building location
january_min_temp:	minimum temperature in January (in **Fahrenheit**) at the location of the building
january_avg_temp:	average temperature in January (in **Fahrenheit**) at the location of the building
january_max_temp:	maximum temperature in January (in **Fahrenheit**) at the location of the building
cooling_degree_days:	Degree days are measures of how cold or warm a location is. A degree day compares the mean outdoor temperatures recorded for a location to a standard temperature, usually 65° Fahrenheit (F) in the United States. The more extreme the outside temperature, the higher the number of degree days. A high number of degree days generally results in higher levels of energy use for space heating or cooling. (CDD) are a measure of how hot the temperature was on a given day or during a period of days.
heating_degree_days:	(HDD) are a measure of how cold the temperature was on a given day or during a period of days.
precipitation_inches:	annual precipitation in **inches** at the location of the building
snowfall_inches:	annual snowfall in **inches** at the location of the building
snowdepth_inches:	annual snow depth in **inches** at the location of the building
avg_temp:	average temperature in **Fahrenheit** over a year at the location of the building
days_below_30F:	total number of days below 30 degrees **Fahrenheit** at the location of the building
days_below_20F:	total number of days below 20 degrees **Fahrenheit** at the location of the building
days_below_10F:	total number of days below 10 degrees **Fahrenheit** at the location of the building
days_below_0F:	total number of days below 0 degrees **Fahrenheit** at the location of the building
days_above_80F:	total number of days above 80 degrees Fahrenheit at the location of the building
days_above_90F:	total number of days above 90 degrees Fahrenheit at the location of the building
days_above_100F:	total number of days above 100 degrees Fahrenheit at the location of the building
days_above_110F:	total number of days above 110 degrees Fahrenheit at the location of the building
direction_max_wind_speed:	wind direction for maximum wind speed at the location of the building. Given in 360-degree compass point directions (e.g. 360 = north, 180 = south, etc.)
direction_peak_wind_speed:	wind direction for peak wind gust speed at the location of the building. Given in 360-degree compass point directions (e.g. 360 = north, 180 = south, etc.)
max_wind_speed:	maximum wind speed at the location of the building
days_with_fog:	number of days with fog at the location of the building
site_eui	Energy Use Intensity (EUI) refers to the amount of energy used of the building per square foot annually. It's calculated by dividing the total energy consumed by the building in a year by the total gross floor area. EUI is the prime indicator of a building's energy performance. Generally, a low EUI signifies good energy performance(But not a norm). EUI is often used to compare buildings of the same use type.

Figure 1 : A detailed description of each feature

Exploratory Data Analysis

The data set is comparatively large and there are 75,757 instances (rows) in the dataset. The dataset has 63 variables(features) which include both categorical and numerical variables.



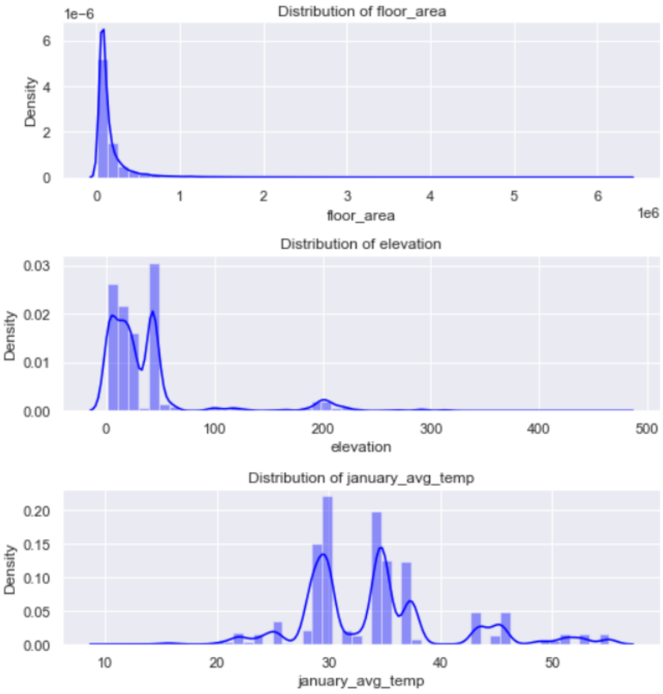
	year_factor	floor_area	year_built	energy_star_rating	elevation	january_min_temp	january_avg_temp	january_max_temp
count	75757.000000	7.575700e+04	73920.000000	49048.000000	75757.000000	75757.000000	75757.000000	75757.000000
mean	4.367755	1.659839e+05	1952.306764	61.048605	39.506323	11.432343	34.310468	59.054952
std	1.471441	2.468758e+05	37.053619	28.663683	60.656596	9.381027	6.996108	5.355458
min	1.000000	9.430000e+02	0.000000	0.000000	-6.400000	-19.000000	10.806452	42.000000
25%	3.000000	6.237900e+04	1927.000000	40.000000	11.900000	6.000000	29.827586	56.000000
50%	5.000000	9.136700e+04	1951.000000	67.000000	25.000000	11.000000	34.451613	59.000000
75%	6.000000	1.660000e+05	1977.000000	85.000000	42.700000	13.000000	37.322581	62.000000
max	6.000000	6.385382e+06	2015.000000	100.000000	1924.500000	49.000000	64.758065	91.000000

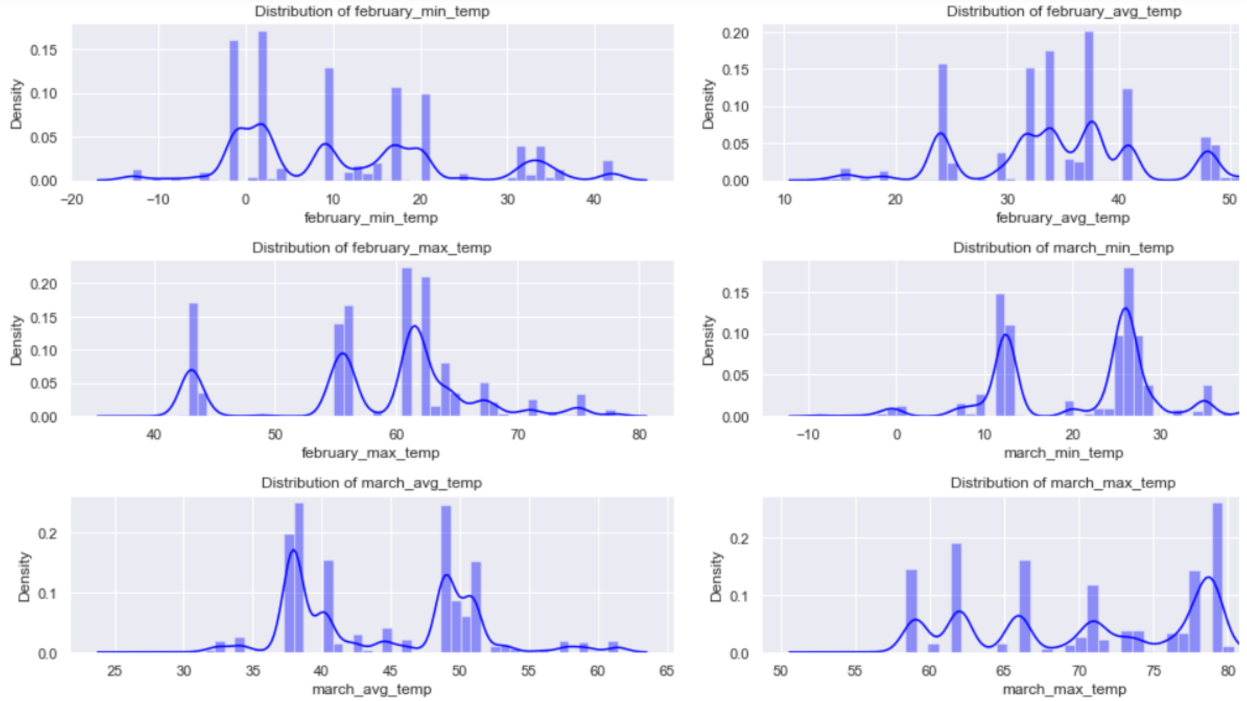
	february_min_temp	february_avg_temp	...	days_below_0F	days_above_80F	days_above_90F	days_above_100F	days_above_110F
count	75757.000000	75757.000000	...	75757.000000	75757.000000	75757.000000	75757.000000	75757.000000
mean	11.720567	35.526837	...	0.876764	82.709809	14.058701	0.279539	0.002442
std	12.577272	8.866697	...	2.894244	25.282913	10.943996	2.252323	0.142140
min	-13.000000	13.250000	...	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	31.625000	...	0.000000	72.000000	6.000000	0.000000	0.000000
50%	9.000000	34.107143	...	0.000000	84.000000	12.000000	0.000000	0.000000
75%	20.000000	40.879310	...	0.000000	97.000000	17.000000	0.000000	0.000000
max	48.000000	65.107143	...	31.000000	260.000000	185.000000	119.000000	16.000000

	direction_max_wind_speed	direction_peak_wind_speed	max_wind_speed	days_with_fog	site_eui
count	34675.000000	33946.000000	34675.000000	29961.000000	75757.000000
mean	66.552675	62.779974	4.190601	109.142051	82.584693
std	131.147834	130.308106	6.458789	50.699751	58.255403
min	1.000000	1.000000	1.000000	12.000000	1.001169
25%	1.000000	1.000000	1.000000	88.000000	54.528601
50%	1.000000	1.000000	1.000000	104.000000	75.293716
75%	1.000000	1.000000	1.000000	131.000000	97.277534
max	360.000000	360.000000	23.300000	311.000000	997.866120

Figure 2: A brief statistical overview of numerical data in the dataset

Distribution of few Numerical Data





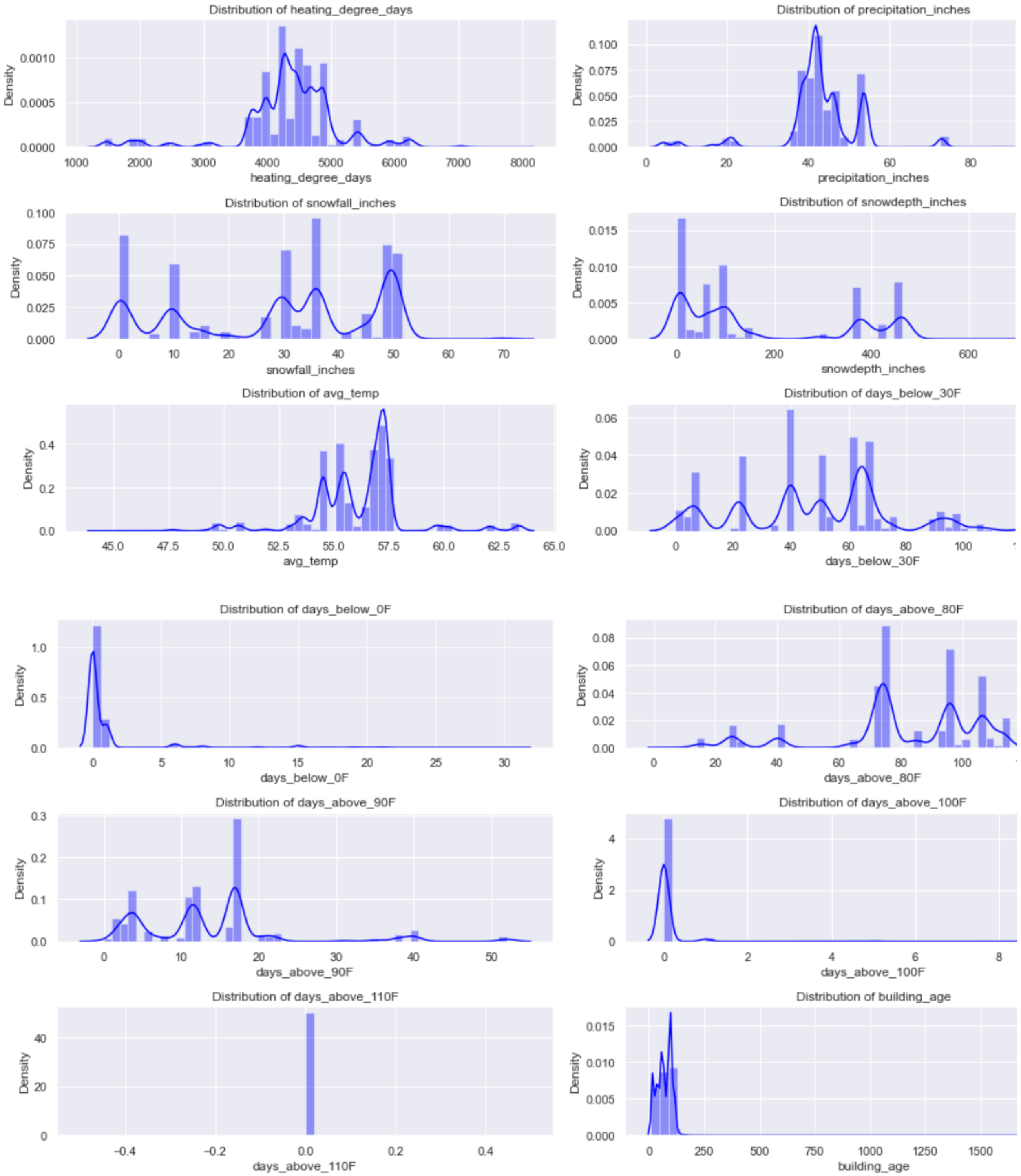
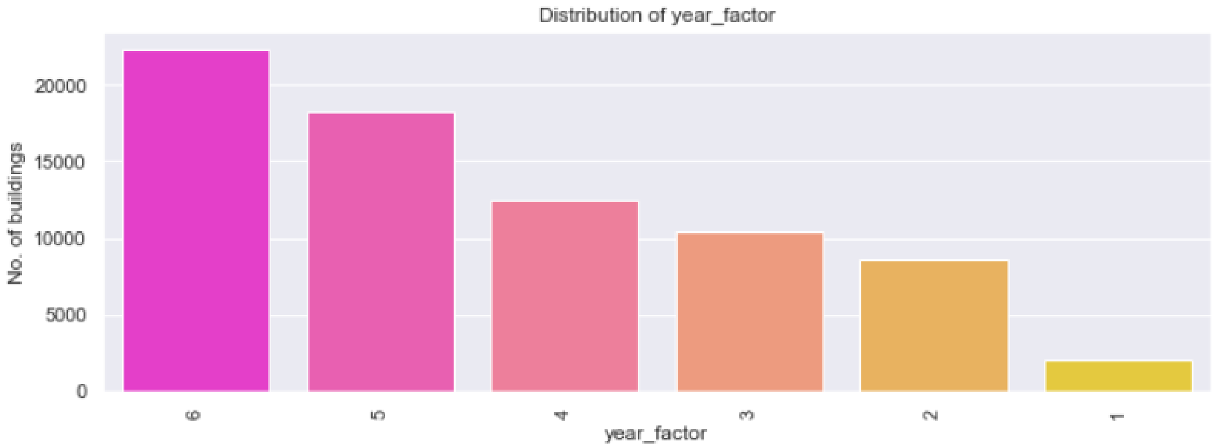
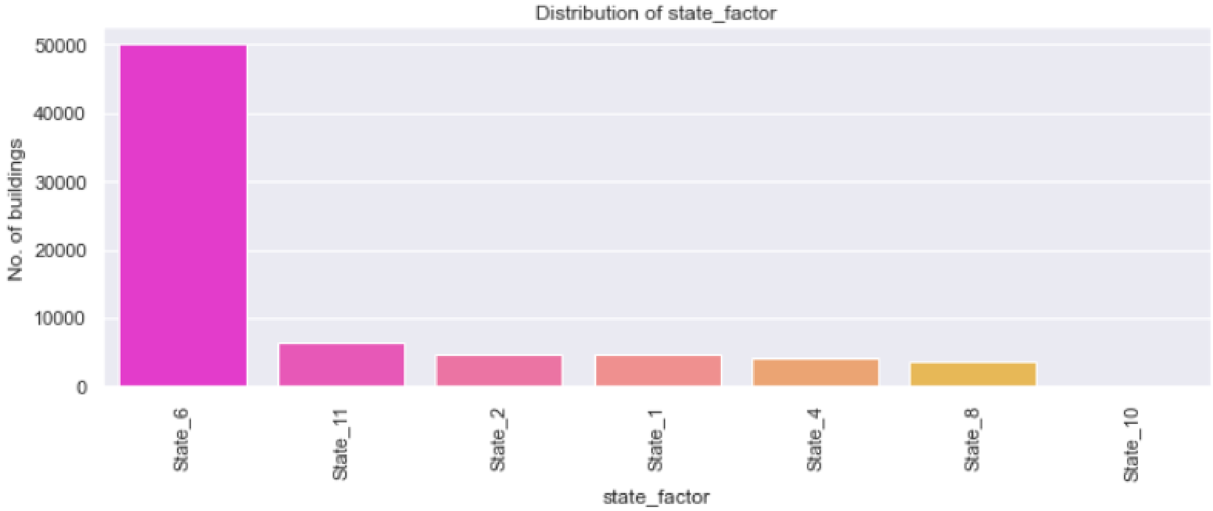


Figure 3: Distribution of few numerical Data

Distribution of few Numerical Data

The categorical data is not equally distribution for all four categorical variables. As an example, the distribution of state_factor shows that it is highly bias to the state_6. And the graph of distribution of facility type shows that the Multifamily_uncategorized category is dominating that variable. Those factors could affect to a machine learning model. However, I did not attempt any adjustment for these data.





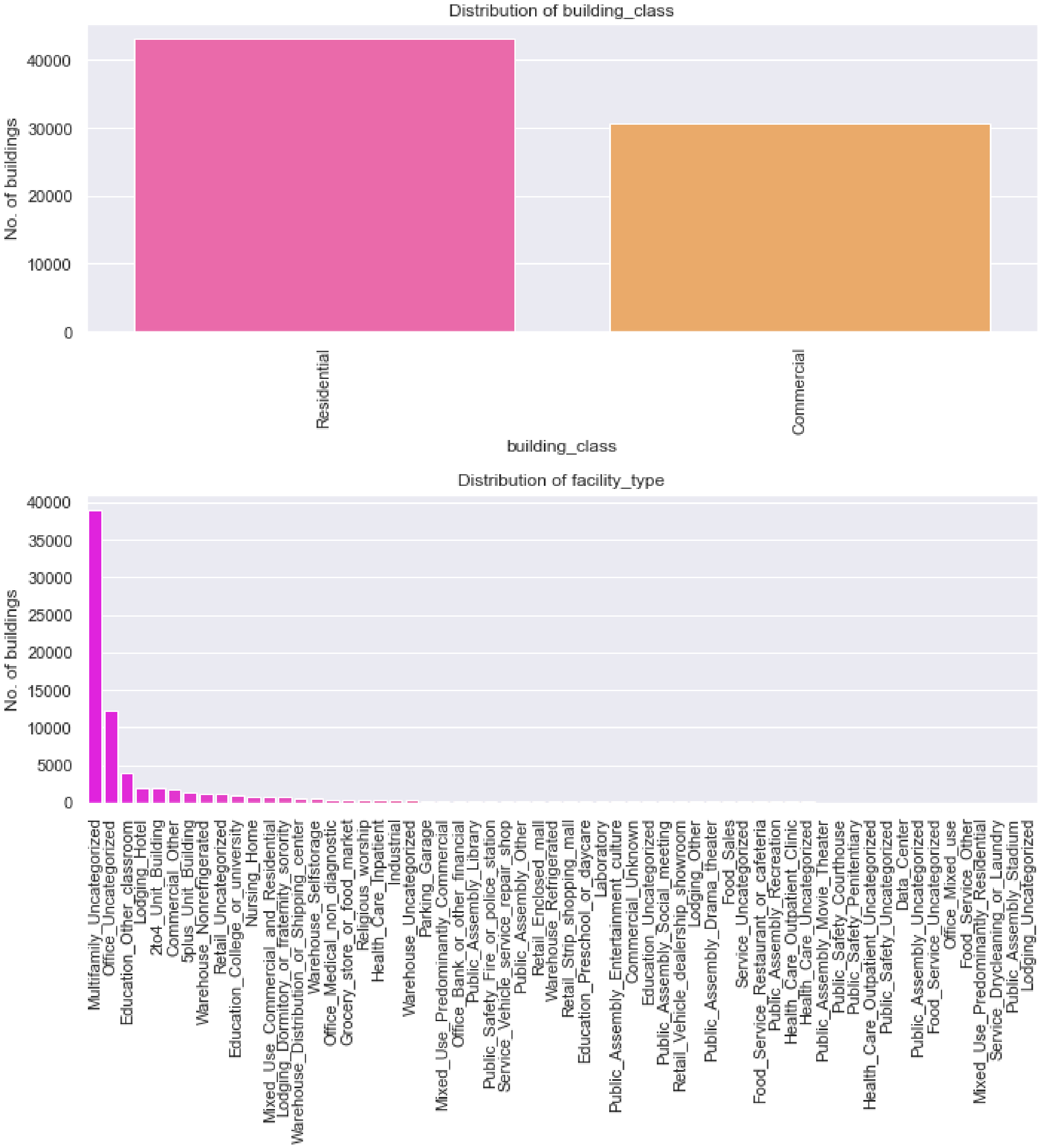


Figure 4: Distribution of categorical Data

The Correlation Statistics of Numerical Variables

Correlation describes the relationship between features. It can be positive; an increase in one feature's value improves the value of the other variable, or negative; an increase in one feature's value decreases the value of the other variable.

The correlation matrix shows a lot of climate features seem to be correlated(positively or negatively) with each other. It is possible to combine some of those features with each other to reduce the number of features and, eventually they could help to increase the accuracy of the model.

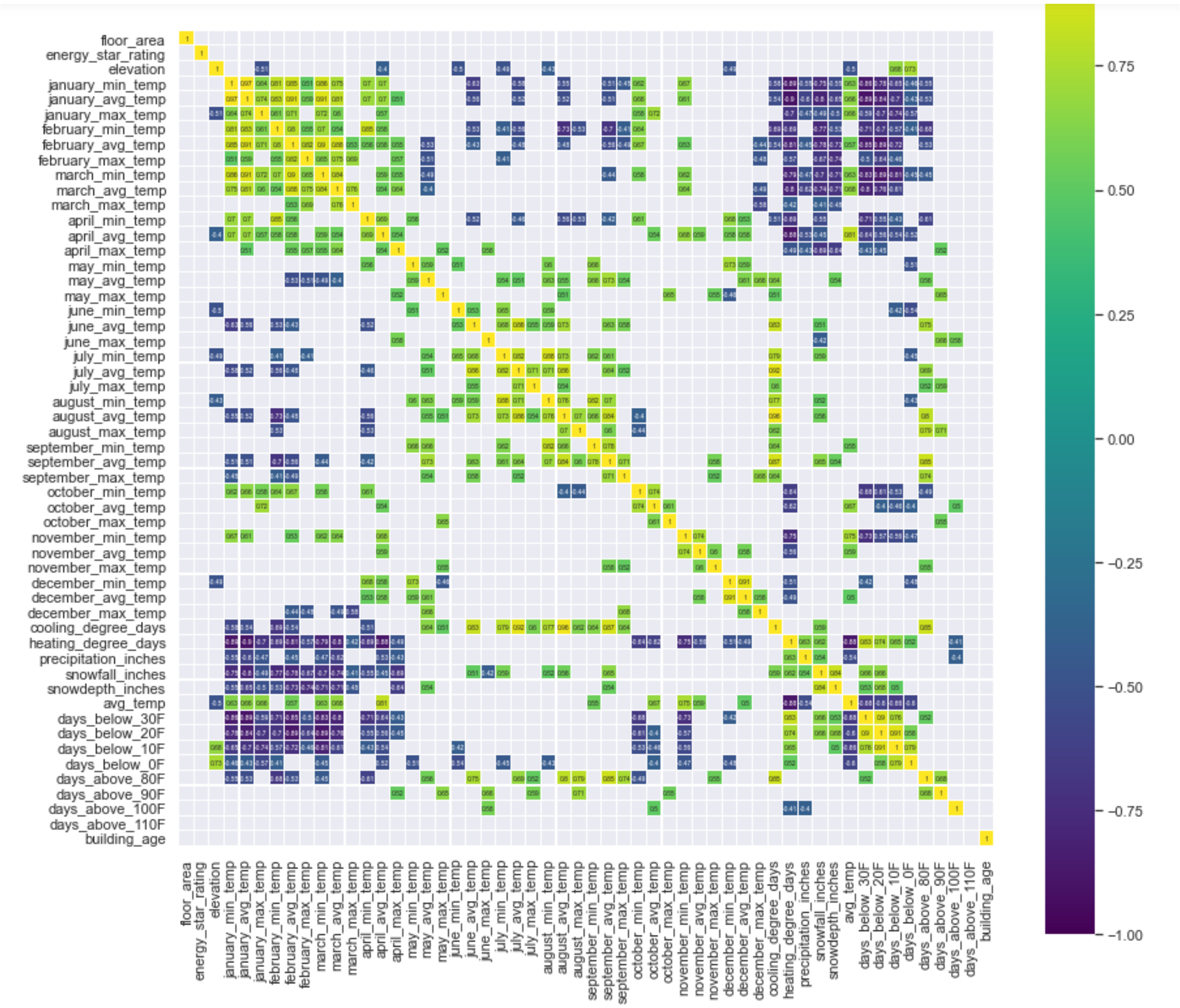
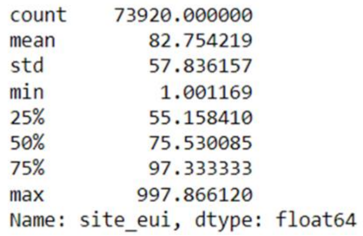
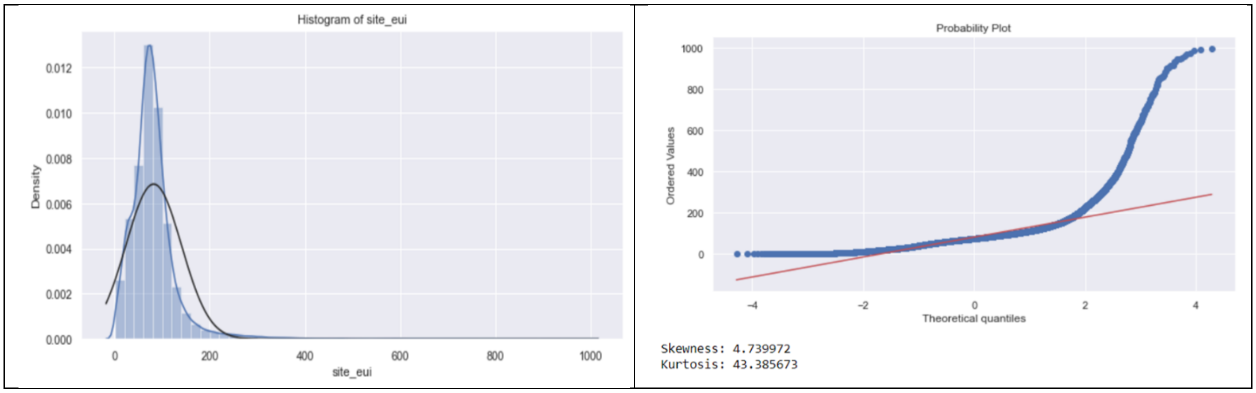


Figure 5: The correlation between numerical variables

Analyze the Dependent Variable 'site_eui'



```
count    73920.000000
mean        82.754219
std         57.836157
min          1.001169
25%         55.158410
50%         75.530085
75%         97.333333
max        997.866120
Name: site_eui, dtype: float64
```



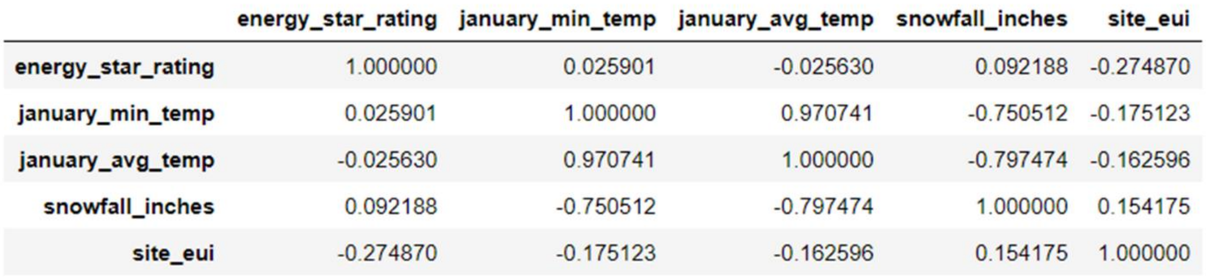
```
Skewness: 4.739972
Kurtosis: 43.385673
```

Figure 6: Analyze the Dependent Variable 'site_eui'

The data has a positive Kurtosis(leptokurtic) and a positively Skewed distribution.

Correlations with the Target Variable

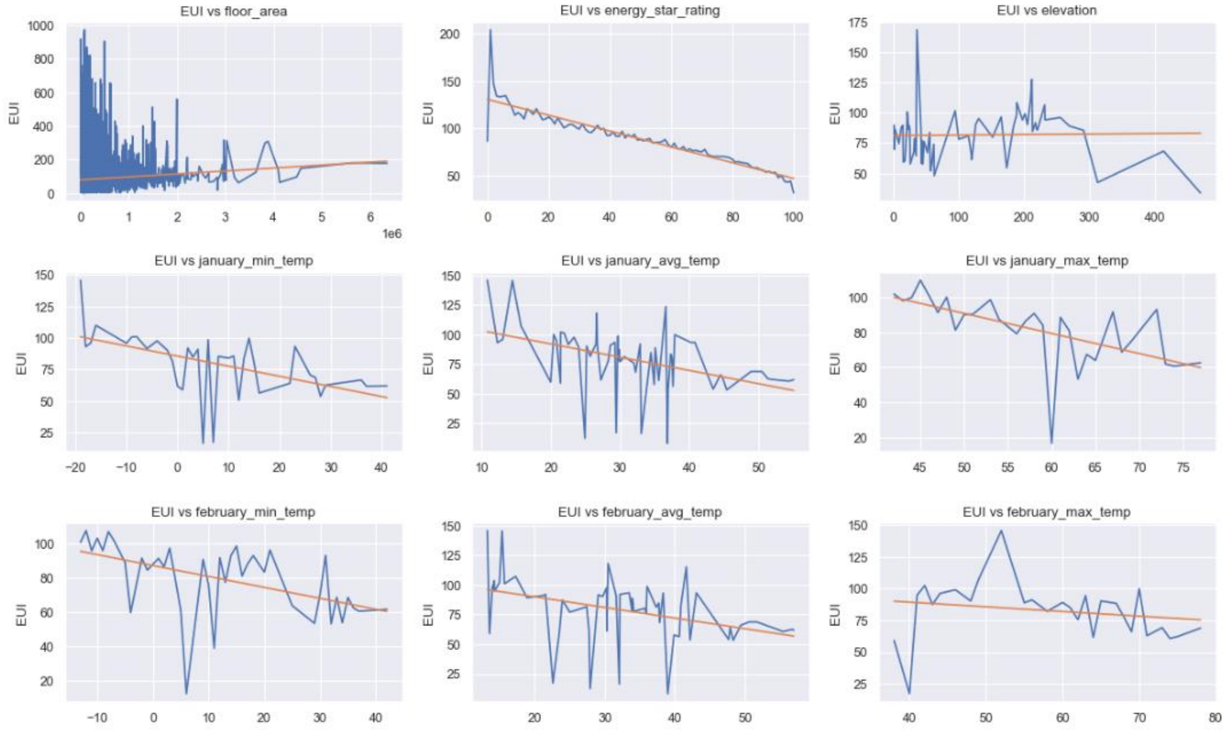
The following table shows the five best correlations with the target variable site_eui. Energy_star_rating shows the highest correlation, and it is a negative correlation. Simply this correlation describes that a building with low site_eui has a high Energy_star_rating and it is an energy efficient building.

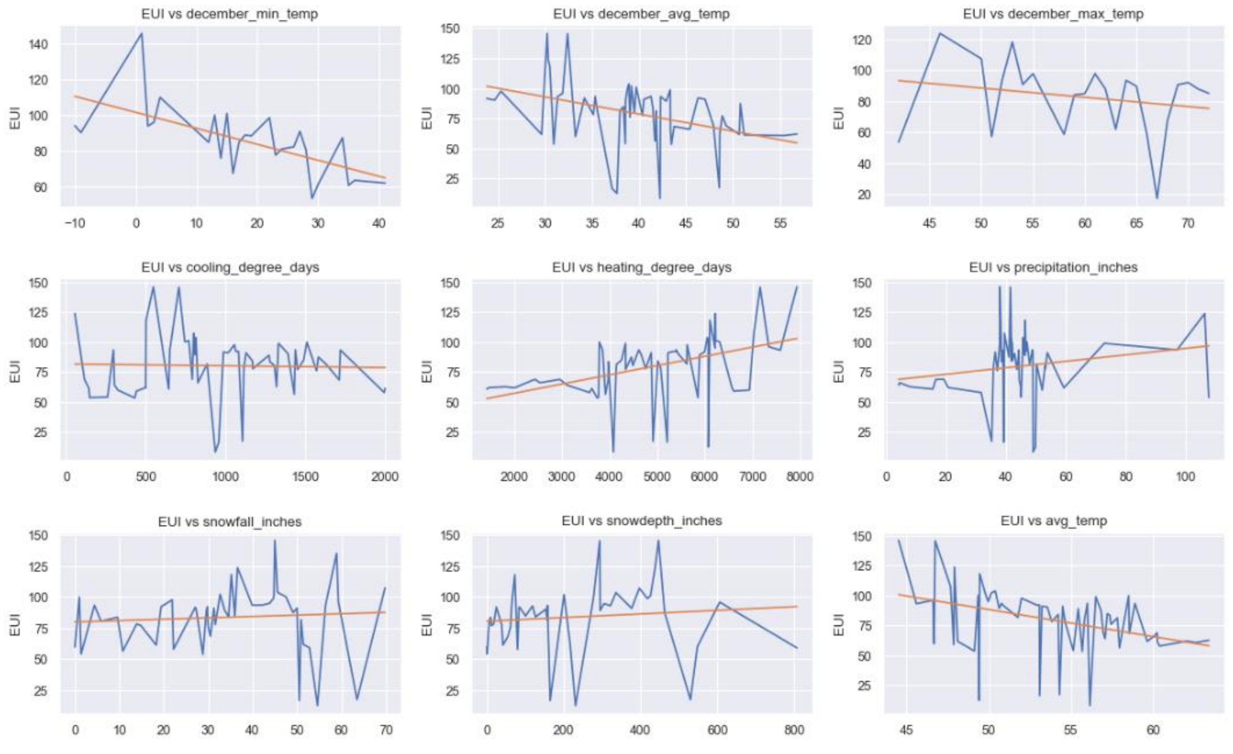


	energy_star_rating	january_min_temp	january_avg_temp	snowfall_inches	site_eui
energy_star_rating	1.000000	0.025901	-0.025630	0.092188	-0.274870
january_min_temp	0.025901	1.000000	0.970741	-0.750512	-0.175123
january_avg_temp	-0.025630	0.970741	1.000000	-0.797474	-0.162596
snowfall_inches	0.092188	-0.750512	-0.797474	1.000000	0.154175
site_eui	-0.274870	-0.175123	-0.162596	0.154175	1.000000

Figure 7: The five best correlations with the target variable

Target variable vs Numeric Features





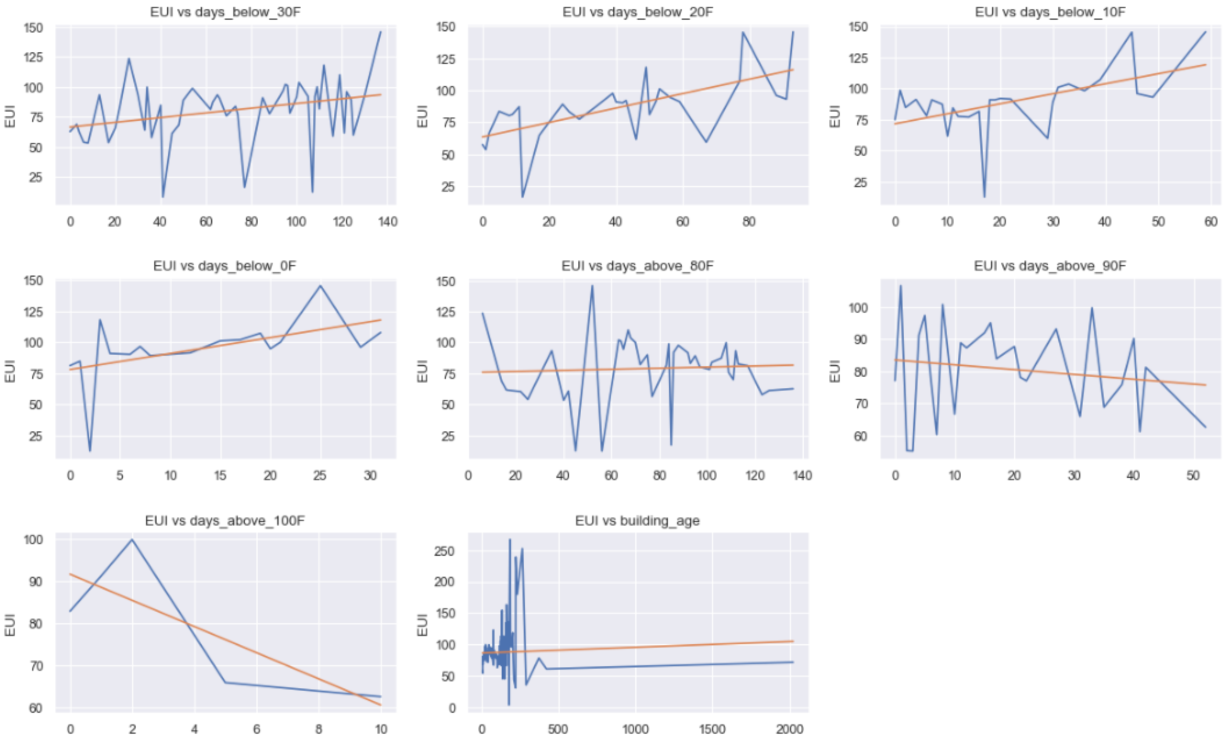
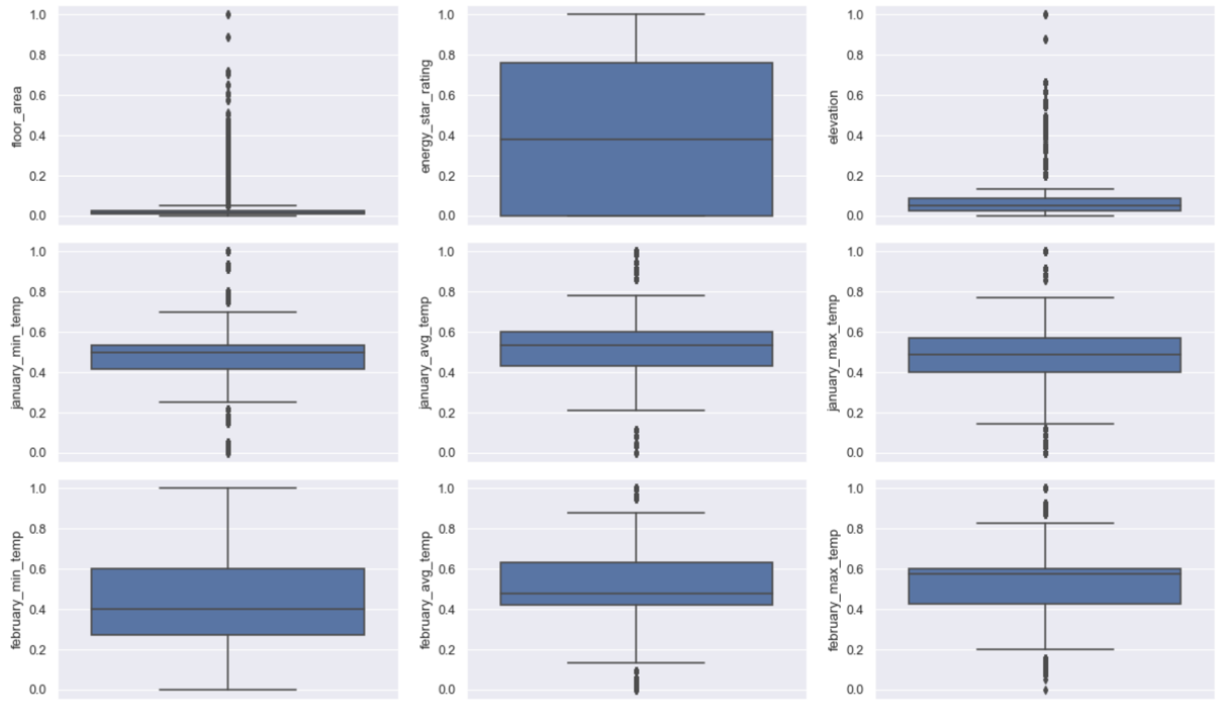
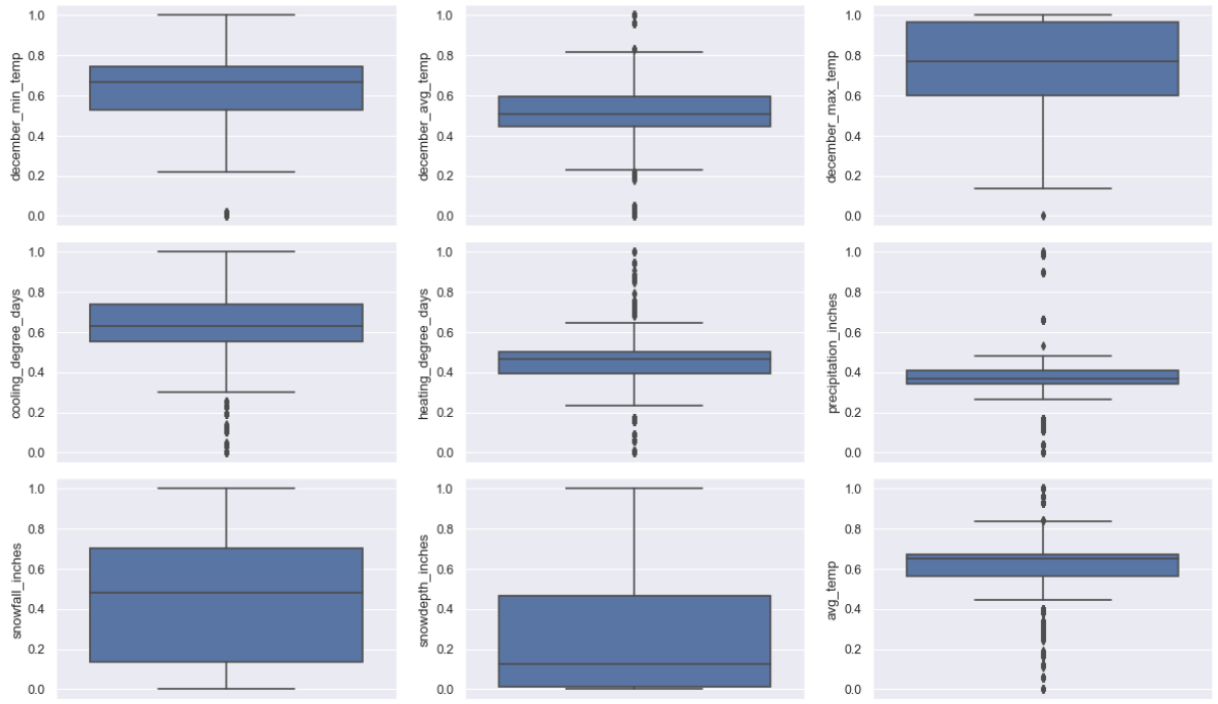


Figure 8: Analysis to see any trend between the numeric feature vs site_eui

Identify Outliers- box plots for numerical attributes


floor_area
energy_star_rating
elevation
january_min_temp
january_avg_temp
january_max_temp
february_min_temp
february_avg_temp
february_max_temp
0.0
0.2
0.4
0.6
0.8
1.0


december_min_temp
december_avg_temp
december_max_temp
cooling_degree_days
heating_degree_days
precipitation_inches
snowfall_inches
snowdepth_inches
avg_temp
0.0
0.2
0.4
0.6
0.8
1.0

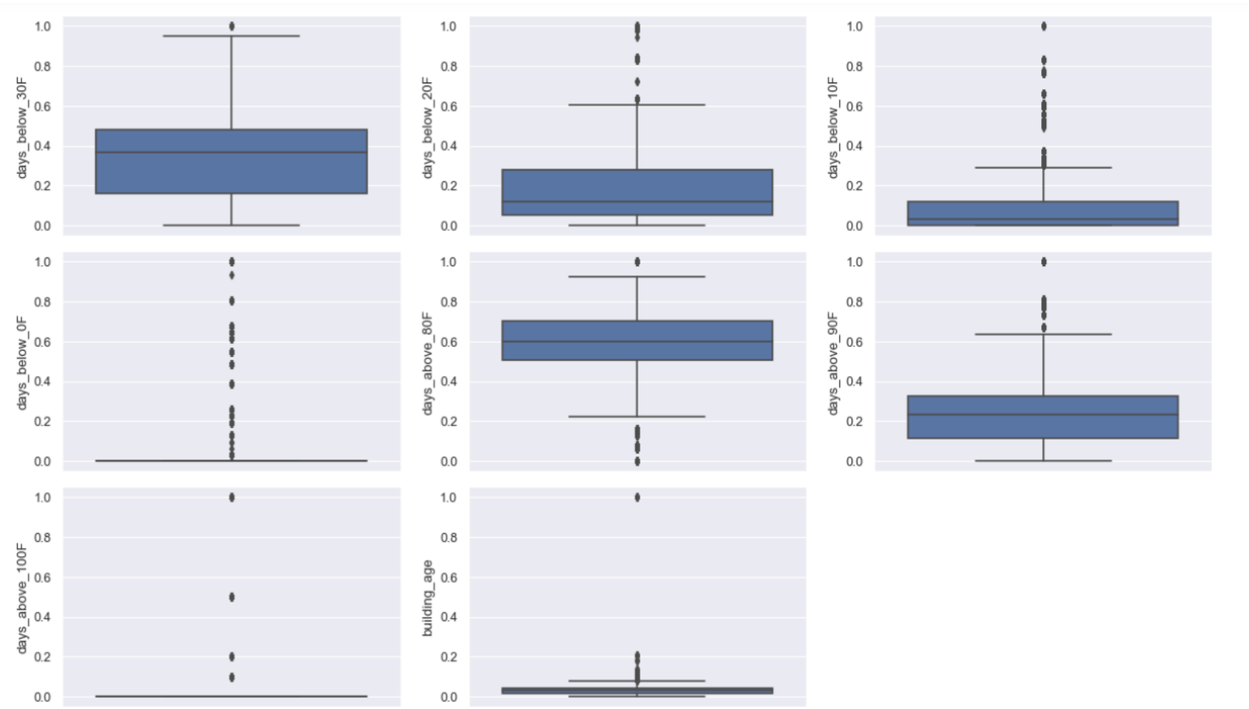


Figure 9 : Box plots for numerical attributes

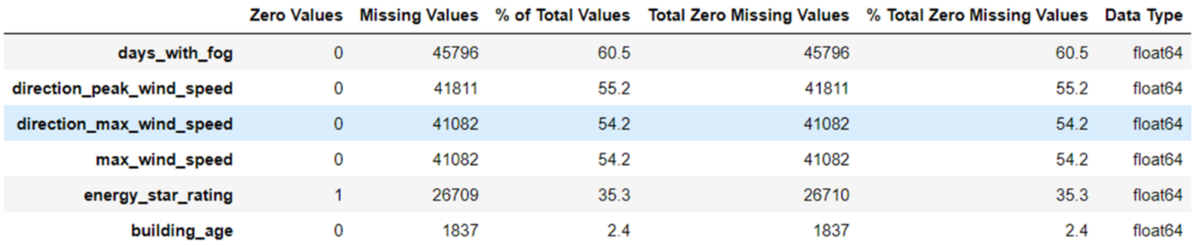
Box plots are useful as they show outliers within a data set. An outlier is an observation that is numerically distant from the rest of the data. When reviewing a box plot, an outlier is defined as a data point that is located outside the whiskers of the box plot. Above graphs shows the box plots of few variables in the dataset. We can see that most of the variables are having outliers. As a best practice in machine learning we need to remove outliers from our data, and it will help to improve the accuracy of the models.

Data Preprocessing

Data preprocessing is an important process in machine learning. Preprocessing is a process of identifying missing values, inconsistencies, and noise of a dataset and take required action to correct them. It can help to improve the quality of the data. Also, data preprocessing can help to reduce the required time and resources to train the machine learning algorithm. Eventually, data preprocessing can help to improve the accuracy of the machine learning algorithm.

Dealing with Missing Data

Missing data is a common issue in most of the raw data sets. This dataset is also having a considerable number of missing values.



	Zero Values	Missing Values	% of Total Values	Total Zero Missing Values	% Total Zero Missing Values	Data Type
days_with_fog	0	45796	60.5	45796	60.5	float64
direction_peak_wind_speed	0	41811	55.2	41811	55.2	float64
direction_max_wind_speed	0	41082	54.2	41082	54.2	float64
max_wind_speed	0	41082	54.2	41082	54.2	float64
energy_star_rating	1	26709	35.3	26710	35.3	float64
building_age	0	1837	2.4	1837	2.4	float64

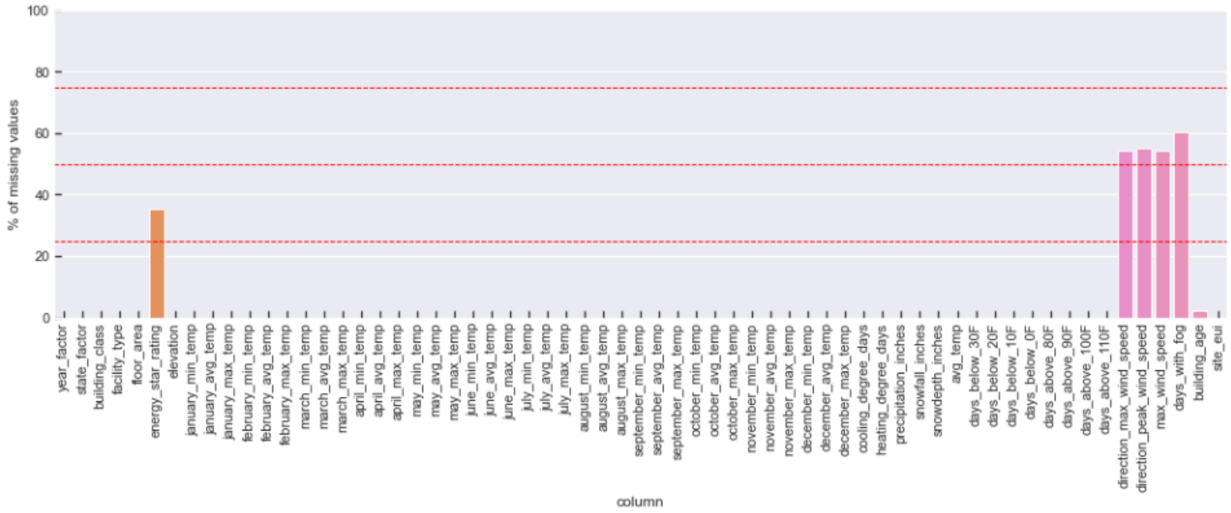


Figure 10: The description of the missing data in the data set.

In my dataset four variables have more than 50% missing values. I just assumed those variables are not important for my analysis and dropped all the columns which contain more than 50% of missing values from the original data frame.

There are 1837 missing values for the variable building age. Since the dataset is considerably large, I decided to drop all the rows which contain missing values for the attribute "building_age“.

Usually, the energy star rating is a score between 1-100. There could be some buildings that do not get a rating yet. So, I replaced all the missing values in "energy_star_rating" with "0".

Encoding Categorical Data

Encoding categorical data is a process of converting categorical data into integer format so that the data with converted categorical values can be provided to the models to give and improve the predictions. Most of the machine learning algorithms expect data in integer format for learning. There are several types of encoding techniques in machine learning. I used Ordinal Encoding.

Since all the categorical variables in the dataset are ordinal, this method fixed the issue. Ordinal encoding converts each label into integer values and the encoded data represents the sequence of labels.

Feature Engineering

We can enhance the dataset by introducing new features into it. They create new source of information that the model can consider and increase the quality of the dataset.

In my data set the attribute "year_building" has no meaning. So, I turned it to a new attribute "building_age“.

Feature Scaling

Feature Scaling should be performed on independent variables that vary in magnitudes, units, and range to standardize to a fixed range. If no scaling, then a machine learning algorithm assign higher weight to greater values regardless of the unit of the values. As the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization.

There are few types of feature scaling available in machine learning and for this project I used Min-Max Scaling. In min-max scaling we subtract the minimum value in the dataset with all the values and then divide this by the range of the dataset(maximum-minimum). And the final dataset will lie between 0 and 1. This technique is also prone to outliers. However, I did not remove outliers from the dataset.

Model Building

Training and Validation Data

All the machine learning algorithms learn from data by finding relationships, developing understanding, making decisions, and building its confidence by using the training data we provide to a machine learning model. A machine learning model will perform based on what training data we have given to a model. In my analysis I divided the cleaned original dataset 80:20 and used 80% of data instances as the training and validation data.

Test Data

A separate unseen test dataset provides a good opportunity for evaluating a model after training and evaluation it with training data. The test set is only used once our machine learning model is trained correctly using the training set. If the predictions that the model makes on the dataset it was trained on are much better than the predictions the model makes on a test dataset that was not seen during training, then the model is likely overfitting.

Model Evaluation Metrics

Quantifying the accuracy of a model is an important step to justifying the usage of the model. One of the simplest methods for calculating the correctness of a model is to use the error between predicted value and actual value. Using this error, we can derive many different metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) or R-Squared Score.

In this project I used RMSE and R-Squared Score for model performance analysis.

RMSE is the square root of the average value of squared error in a set of predicted values, without considering direction. It ranges from 0 to infinity. Lower weight shows a better model. If the model has large errors, it gives a higher weight for RMSE.

Coefficient of Determination (R-squared) provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.

Model Training, Evaluation and Testing

There are different types of regression algorithms are available in machine learning. Regression means to predict the value using the input data. Regression models are mostly used to find the relationship between the variables and forecasting. They differ based on the kind of relationship between dependent and independent variables.

In this project I used following regression algorithms for model building.

Simple Linear Regression: Simple linear regression is a target variable based on the independent variables. Linear regression is a machine learning algorithm based on supervised learning which performs the regression task.

Support Vector Regression: Support vector regression identifies a hyperplane with the maximum margin such that the maximum number of data points is within the margin. Because of the time and resource handling difficulty, I could not finish Support Vector Regression in my project.

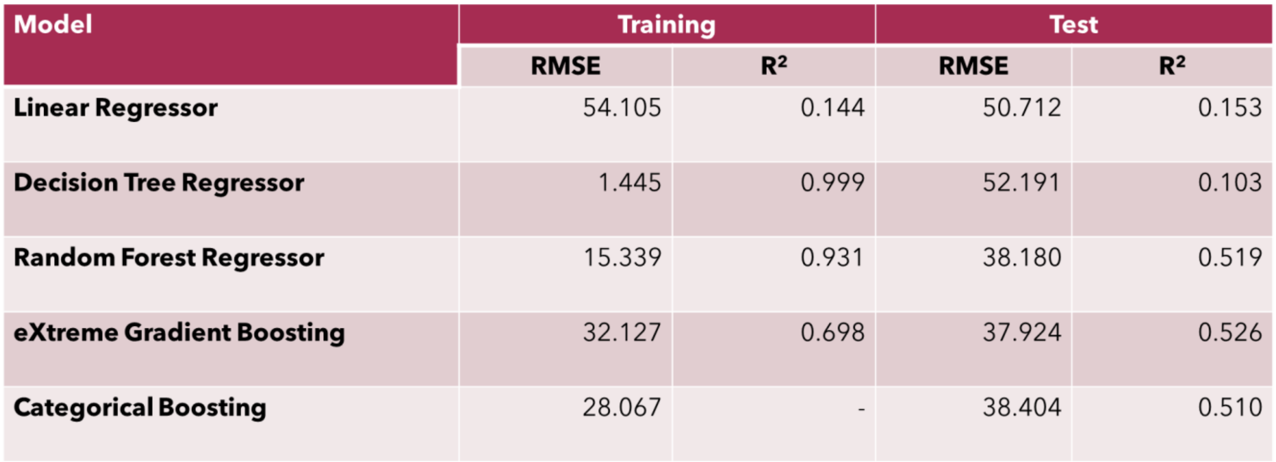
Decision Tree Regression: The decision tree is a tree that is built by partitioning the data into subsets containing instances with similar values. It can use for regression and classification also.

Random Forest Regression: Random Forest is an ensemble approach where we take into account the predictions of several decision regression trees.

Extreme gradient boosting or XGBoost: XGBoost is an implementation of gradient boosting that’s designed for computational speed and scale. XGBoost leverages multiple cores on the CPU, allowing for learning to occur in parallel during training.

CatBoost or Categorical Boosting: CatBoost is an open-source boosting library developed by Yandex. Unlike other gradient boosting algorithms (require numeric data), CatBoost automatically handles categorical features.

First, I did the regression by using simple base models build with the regression algorithms and obtained following results.



Model	Training		Test	
	RMSE	R²	RMSE	R²
Linear Regressor	54.105	0.144	50.712	0.153
Decision Tree Regressor	1.445	0.999	52.191	0.103
Random Forest Regressor	15.339	0.931	38.180	0.519
eXtreme Gradient Boosting	32.127	0.698	37.924	0.526
Categorical Boosting	28.067	-	38.404	0.510

Figure 11: Base model performances

Linear regression tries to capture the linear relationship between the input variables and the target variable. The above results show that the error weights are very high and model fitting goes bad in linear regression. So, we can say that there are not any linier relationships between our input variables and target variables.

The Decision Tree Regression and Random Forest Regression show excellent performance for training data. But the models are not working well for the test dataset. So, we can assume that these models are overfitting.

Extreme gradient boosting and Categorical Boosting models show considerably good performances. However, we cannot assume as those models are not overfitting by looking at these results.

Model Improving Techniques

A machine learning process involves training different models on the dataset and selecting the one with best performance. As shown in above, evaluating the performance of algorithm is not always a straightforward task. There are several factors that can help you determine which algorithm performance best. One such factor is the performance on cross validation set and another other factor is the choice of parameters for an algorithm.

Cross Validation

Normally, in machine learning process, we divide data into training and test sets. The training set is then used to train the model and the test set is used to evaluate the performance of the model. However, this approach may lead to variance problems. Simply, the accuracy obtained on one test can highly deviate from the accuracy got on another test set using the same algorithm.

To overcome this issue, we can use K-Fold Cross-Validation for performance evaluation where K is any number. In K-Fold Cross-Validation we divide the data into K folds. Out of the K folds, K-1 sets are used for training while the remaining set is used for testing. The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set. Finally, we can in fact find the variance in the overall result by using standard deviation of the results obtained from each fold. This is helping to minimize the variance problem in our model performance evaluation.

Grid Search

A Machine Learning model is a mathematical model with various parameters that need to be learned from the data. When we are training a model, these parameters are fitting with existing data.

The other kind of parameters in a machine learning model are known as Hyperparameters. These parameters express important properties of the model such as its complexity or how fast it should learn. They cannot learn directly from a regular training process. We usually fix them before the actual training process begins. Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. The two widely use strategies for Hyperparameter tuning are Grid Search Cross Validation(GridSearchCV) and Randomized Search Cross Validation(RandomizedSearchCV). In this project I used GridSearchCV Hyperparameter tuning for this project.

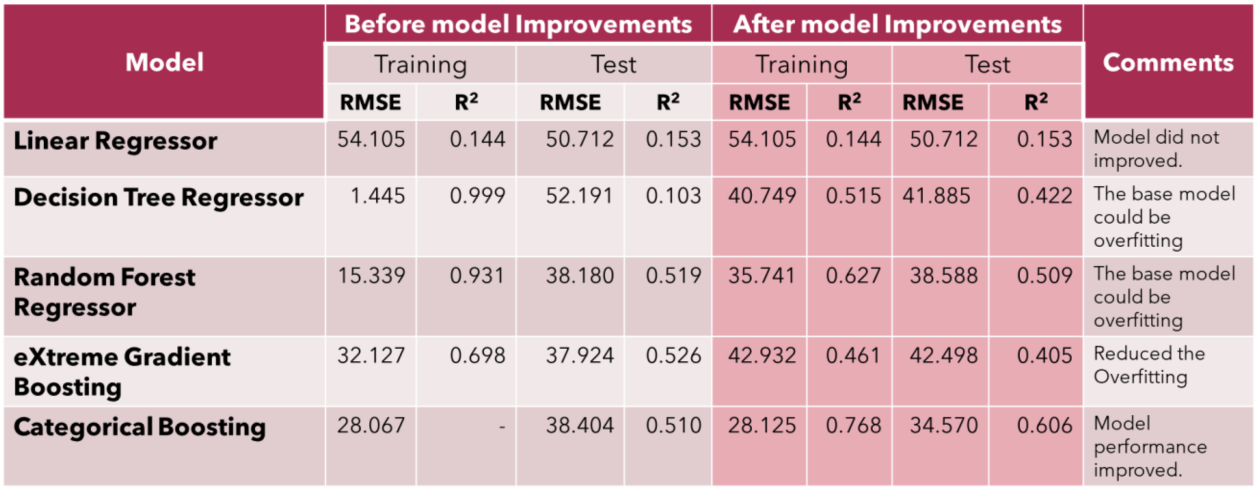
Feature Selection

We called input variables we give to a machine learning models as features. Each column in a dataset represents a feature. To train an optimal model, we need to make sure that we use only the essential features. Having too many features could be a disadvantage for a machine learning model. They could lead to capture the unimportant patterns and learn from noise. Therefore, it is important to find out the optimal set of parameters for a best model performance. The method of choosing the important parameters of a dataset is called Feature Selection. Feature selection could help machine learning algorithms to run more efficiently (less space or time complexity) and be more effective.

In this project, I used only supervised learning machine learning algorithms. There are three main types of feature selection models in supervised learning called: Filter Method, Wrapper Method, and Intrinsic Method.

I selected the Recursive Feature Elimination (RFE) in this project. It is a wrapper-type feature selection algorithm. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains.

Model Performance Evaluation after Applying Model Improving Techniques



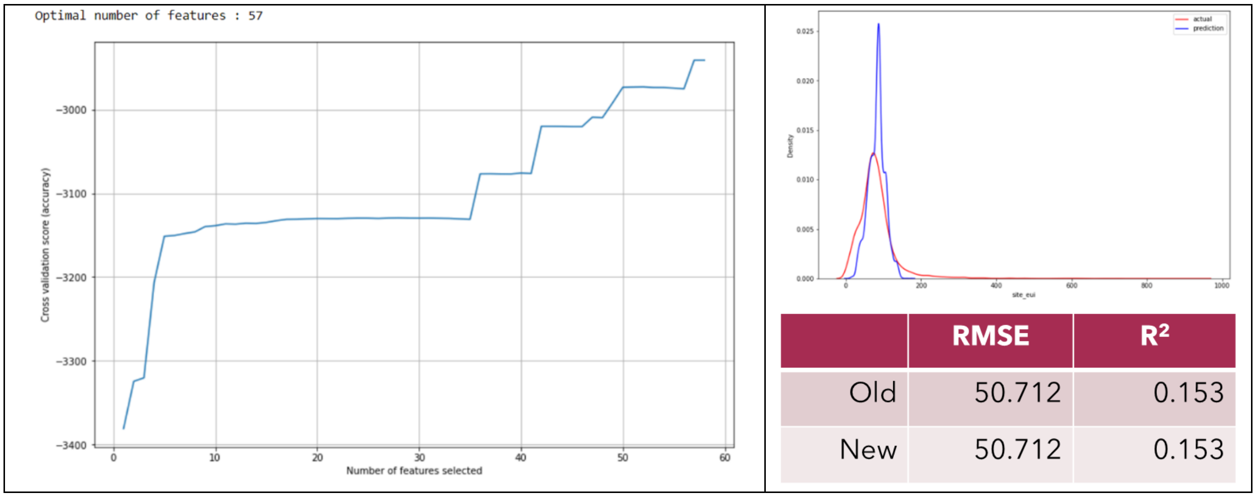
Model	Before model Improvements				After model Improvements				Comments
	Training		Test		Training		Test		
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	
Linear Regressor	54.105	0.144	50.712	0.153	54.105	0.144	50.712	0.153	Model did not improved.
Decision Tree Regressor	1.445	0.999	52.191	0.103	40.749	0.515	41.885	0.422	The base model could be overfitting
Random Forest Regressor	15.339	0.931	38.180	0.519	35.741	0.627	38.588	0.509	The base model could be overfitting
eXtreme Gradient Boosting	32.127	0.698	37.924	0.526	42.932	0.461	42.498	0.405	Reduced the Overfitting
Categorical Boosting	28.067	-	38.404	0.510	28.125	0.768	34.570	0.606	Model performance improved.

Figure 12: Improved model performances

The results table shows that the Linear Regression model performances have not improved even after performing model improvements. All the other models improved considerably after applying hyperparameter tuning, cross-validation and feature selection techniques.

Feature Selection and Model Improvements

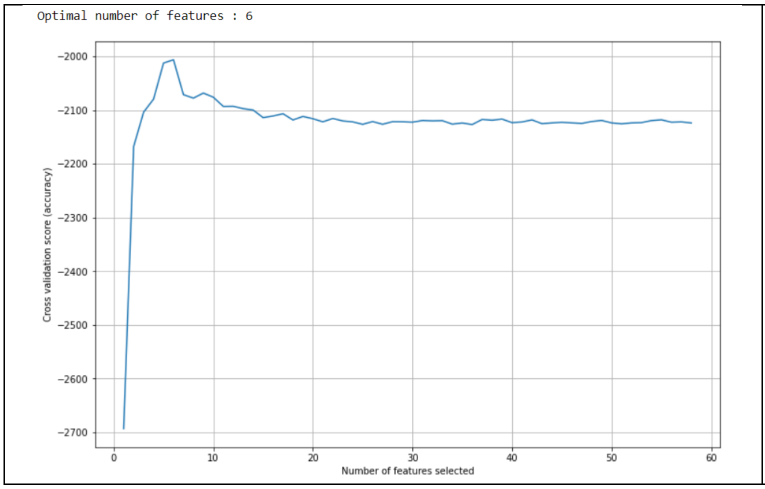
Linear Regression

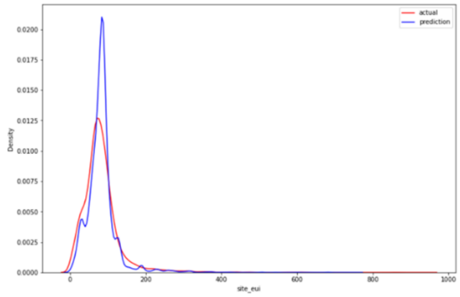


	RMSE	R^2
Old	50.712	0.153
New	50.712	0.153

The Linear Regression model shows the best performance with 57 features. The model is considering only the 'days_above_110F' is not important. However, the accuracy of the base model has not improved.

Decision Tree Regression

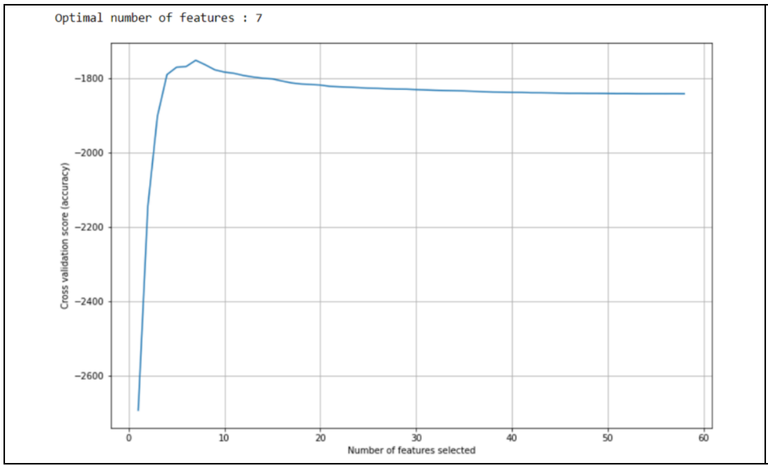


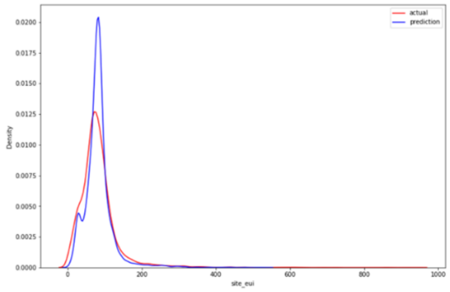


	RMSE	R²
Old	52.191	0.103
New	41.885	0.422

The Decision Tree Regression model shows the best performance with only 6 features. They are 'state_factor', 'facility_type', 'building_class', 'floor_area', 'energy_star_rating', and 'building_age'. The accuracy of the base model has improved considerably after doing hyperparameter tuning, cross validation and Recursive Feature Elimination.

Random Forest Regression

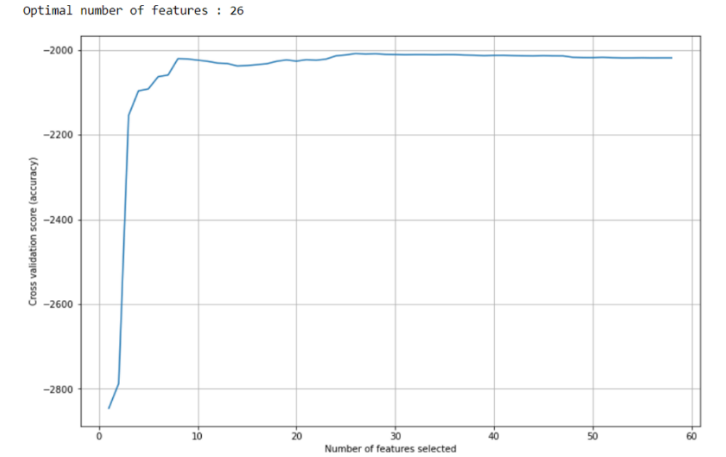


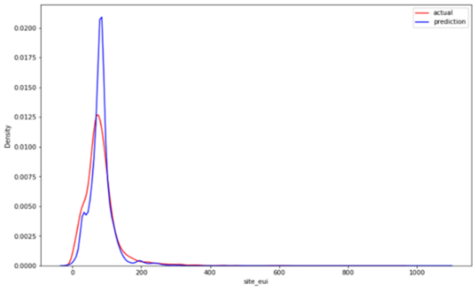


	RMSE	R²
Old	38.180	0.519
New	38.588	0.509

The Random Forest Regression model shows the best performance with 7 features. They are 'state_factor', 'building_class', 'facility_type', 'floor_area', 'energy_star_rating', 'snowdepth_inches', and 'building_age'. The training and evaluation results along with the testing results are showing that the model improved and reduce the overfitting issue considerably after doing hyperparameter tuning, cross-validation and Recursive Feature Elimination.

eXtreme Gradient Boosting

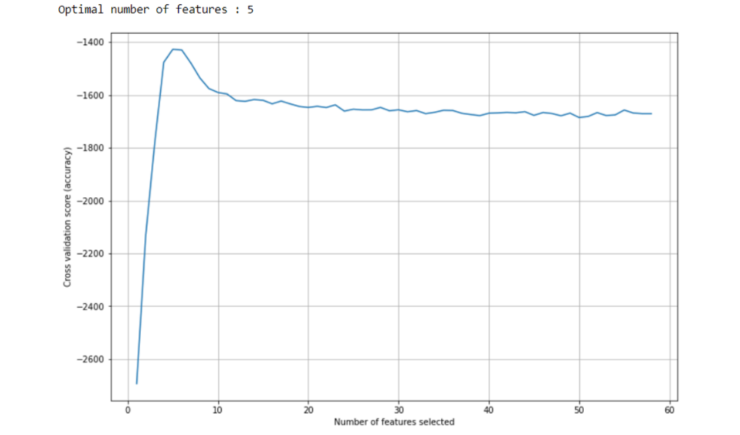


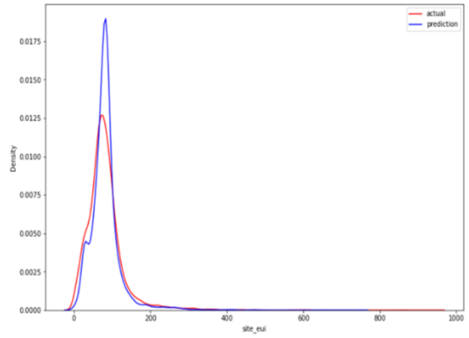


	RMSE	R^2
Old	37.924	0.526
New	42.498	0.405

The eXtreme Gradient Boosting Regression model shows the best performance with 26 features. They are 'facility_type', 'state_factor', 'building_class', 'floor_area', 'energy_star_rating', 'elevation', 'january_min_temp', 'january_avg_temp', 'january_max_temp', 'february_avg_temp', 'february_max_temp', 'march_max_temp', 'april_avg_temp', 'may_avg_temp', 'june_min_temp', 'july_min_temp', 'july_avg_temp', 'september_max_temp', 'october_max_temp', 'december_max_temp', 'heating_degree_days', 'precipitation_inches', 'snowfall_inches', 'snowdepth_inches', 'days_above_100F', and 'building_age'. At first glance, we cannot see any model performance improvement. But we can say that the application of model improving techniques helps to reduce the overfitting issue and improve the stability of the model.

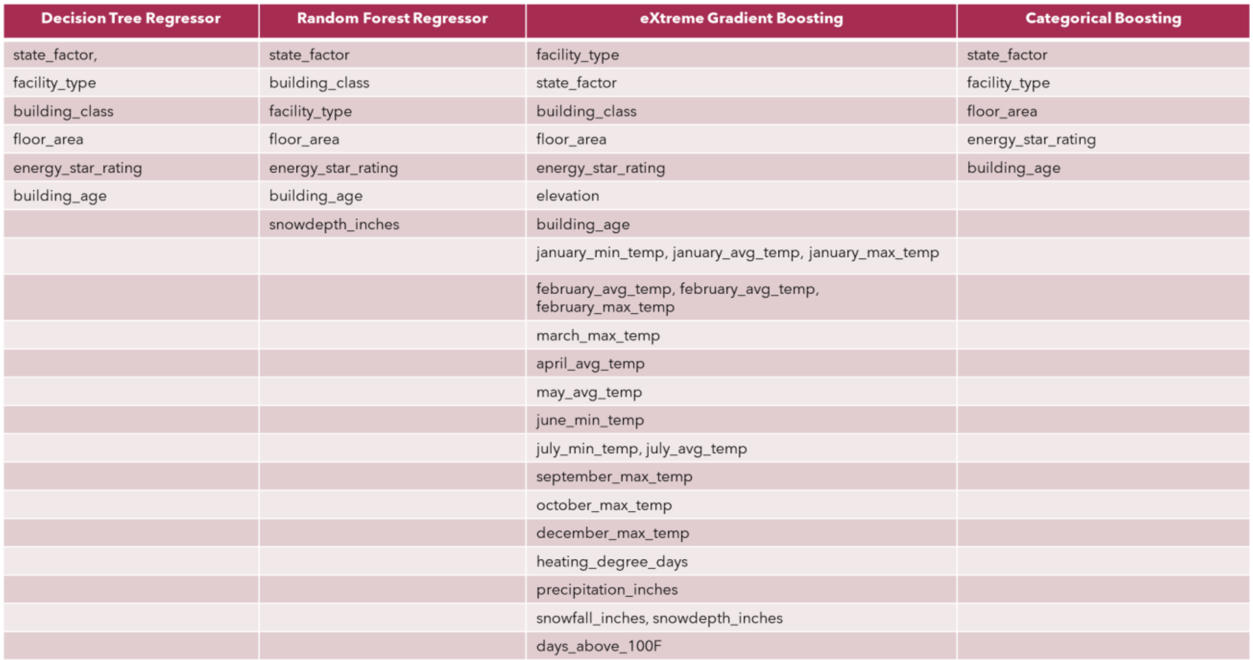
Categorical Boosting





	RMSE	R^2
Old	38.404	0.510
New	34.570	0.606

The Categorical Boosting Regression model shows the best performance with 5 features. They are 'state_factor', 'facility_type', 'floor_area', 'energy_star_rating', and 'building_age'. The accuracy of the base model has improved considerably.



Decision Tree Regressor	Random Forest Regressor	eXtreme Gradient Boosting	Categorical Boosting
state_factor,	state_factor	facility_type	state_factor
facility_type	building_class	state_factor	facility_type
building_class	facility_type	building_class	floor_area
floor_area	floor_area	floor_area	energy_star_rating
energy_star_rating	energy_star_rating	energy_star_rating	building_age
building_age	building_age	elevation	
	snowdepth_inches	building_age	
		january_min_temp, january_avg_temp, january_max_temp	
		february_avg_temp, february_avg_temp, february_max_temp	
		march_max_temp	
		april_avg_temp	
		may_avg_temp	
		june_min_temp	
		july_min_temp, july_avg_temp	
		september_max_temp	
		october_max_temp	
		december_max_temp	
		heating_degree_days	
		precipitation_inches	
		snowfall_inches, snowdepth_inches	
		days_above_100F	

Figure 13: Selected Features from 4 different model implementations

Results Discussion

Categorical Boosting model shows the best performance in site EUI prediction. The RMSE is 34.570 and R2 is 0.606. However, still, we cannot assume this prediction is perfect and the model is not overfitting.

The feature correlation matrix shows a considerable correlation between most climate features. Those correlations can lead to poor performance of regression and other ML algorithms. Those highly correlated features may mislead our models. We can create new features by using those features.

All the selected models considered state_factor, facility_type, building_class, floor_area, energy_star_rating, building_age as important features. When we look at the feature correlation matrix, the above features show a very low correlation with other features. That could be the reason for this feature selection.

According to my view eXtreme Gradient Boosting model is giving the best fit. However, the model is still giving low performances than other models.

I did not remove outliers from feature data. We can remove outliers from the data set and check the model improvements.

By looking at all results, I can say that the model improvement techniques lead to a good model fit and better prediction. But most of those techniques are highly time-consuming and it is the main drawback in this process.