STAT 447B Group Report

Predicting Site Energy Consumption Using Climate Variables and Building Characteristics

By Anjali Chauhan, Sumit Meghlani, Idris Hedayat, Sameer Shankar

March 31, 2022

1. Summary

In this analysis, we sought to develop a model that predicts a site's energy consumption by studying the effects of climate variables and building characteristics on the Site EUI (Energy Usage Intensity) to provide crucial information for potential optimized energy retrofitting. This helps in improving a building's asset performance (utilities). Retrofitting offers a potential upside in the overall performance of the building through improved energy efficiency, increased staff productivity, reduced maintenance costs, and better thermal comfort. By training an ensemble of prediction models like XGBoost, Gradient Boosting Machine, Light GBM, and Support Vector Machine with important variables selected using variable importance method based on Random Forest Regressor, `boruta` package, and Correlation Analysis, we aim to make accurate Site EUI predictions. We achieved a high prediction performance, with the lowest Root Mean Square Error (RMSE) of 18.501 along with a 50% Interval Score of 44.033, Average Interval Length of 17.292 and a coverage rate of 0.493 for the Light GBM method. Whereas we achieved a 80% Interval Score of 68.963, Average Interval Length of 40.015 and a coverage rate of 0.790.

2. Introduction

Climate change is an urgent, and multi-faceted issue heavily impacted by infrastructure. Addressing climate change involves mitigation of Greenhouse Gas (GHG) emissions via changes to electricity systems, transportation, buildings, industry, and land use. According to a report issued by the International Energy Agency, the life cycle of buildings from construction to demolition were responsible for 37% of global energy-related CO_2 emissions in 2020. Yet it is possible to drastically reduce the energy consumption of buildings. For example, retrofitted buildings can reduce heating and cooling energy requirements by 50-90%. Therefore, it is important to optimize energy retrofitting by accurately predicting Site EUI of these buildings. The study aims to investigate and understand the effect of climate variables and building characteristics on the building's EUI (Energy Usage Intensity).

The analysis aims to address the following questions:

- Do the effects of building characteristics outweigh the effect of climate variables on Site EUI and vice-versa?
- Is there a need to build separate models for different Facility types or Building classes?

More specifically, the analysis has the following objectives to answer the questions above:

- To find important climate variable(s) and/or building characteristics variable(s) and determine the effects of said variable(s) on the site EUI
- To translate the relationship between the response and multiple covariates into insightful visualizations

- To model and predict the site EUI values

This report summarizes all of the primary statistical modelling and analysis results associated with the study. The remainder of the report is organized as follows: Section 3 describes the data collection, provides measurement of the variables and summarizes the data. Section 4 presents the data pre-processing and statistical modelling techniques used to answer the aforementioned research questions. Section 5 summarizes and interprets the results of the statistical analysis conducted. Appendices are provided for further exploratory data analysis along with the code used for the statistical modelling. Lastly, Section 6 presents the limitations and challenges in conducting this analysis and Section 7 and 8 cover the conclusion of this study and the next steps for future analysis, respectively.

3. Data

3.1 Description

The data was collected in collaboration with Climate Change AI (CCAI) and Lawrence Berkeley National Laboratory (Berkeley Lab). Data contains roughly 100,000 observations of building energy usage records collected over 7 years, from a number of states within the United States. The dataset consists of building characteristics (e.g. floor area, facility type, etc), and weather data for the building location (e.g. annual average temperature, annual total precipitation, etc), as well as the energy usage for the building (Site EUI). Each row in the data corresponds to the a single building observed in a given year. There are 75757 rows and 64 columns with 3845 outliers and 64448 rows containing at least one column with missing value.

Table 1: Description of Building Characteristics Variables Used for Analysis

	Variable	**Unit**	**Description**
1.	***(Response)*** Site EUI	$kBtu\ ft^{-2}$	The amount of heat and electricity consumed by a building as reflected in utility bills
2.	Energy Star Rating	-	Measure with a score between 1-100 where a higher energy rating means that the building performs better
3.	Floor Area	ft^2	Floor area of the building
4.	Year Built	-	Anonymized year in which the weather and energy usage factors were observed
5.	Elevation	ft	Elevation of the building location
6.	Facility Type	-	Building usage type
7.	Building Class	-	Building classification (residential/commercial)
8.	State Factor	-	Anonymized state in which the building is located
9.	Year Factor	-	Anonymized year in which the weather and energy usage factors were observed

Table 2: Description of Climate Variables Used for Analysis

	Variable	**Unit**	**Description**
1.	[month]'s Minimum Temperature	$°F$	Minimum temperature in [month] at the building location where month from January to December **(12 variables)**
2.	[month]'s Average Temperature	$°F$	Average temperature in [month] at the building location where month from January to December **(12 variables)**
3.	[month]'s Maximum Temperature	$°F$	Maximum temperature in [month] at the building location where month from January to December **(12 variables)**

	Variable	Unit	Description
4.	Cooling Days	-	The number of degrees where the daily average temperature exceeds 65 $°F$
5.	Heating Days	-	The number of degrees where the daily average temperature falls under 65 $°F$
6.	Precipitation	*inches*	Annual precipitation at the building location
7.	Snowfall	*inches*	Annual snowfall at the building location
8.	Average Temperature	$°F$	Average temperature over a year at the building location
9.	Days Below 30F	-	Total number of days below 30 $°F$ at the building location
10.	Days Above 80F	-	Total number of days above 80 $°F$ at the building location
11.	Direction of Max Wind Speed	°	Wind direction for maximum wind speed at the building location
12.	Direction of Peak Wind Speed	°	Wind direction for peak wind gust speed at the building location
13.	Max Wind Speed	ms^{-1}	Maximum wind speed at the building location
14.	Days With Fog	-	Number of days with fog at the building location

Table 3: Summary Statistics of All Climate Variables

Var	site_eui	energy_star_rating	floor_area	year_built	ELEVATION	Year_Factor
Count	75757	49048	75757	73920	75757	75757
Std	58.26	28.66	246875.8	37.05	60.66	1.47
Min	1.00	0.00	943	0.00	-6.40	1.00
25%	54.53	40.00	62379	1927.00	11.90	3.00
50%	75.29	67.00	91367	1951.00	25.00	5.00
Mean	82.58	61.05	165984	1952.31	39.51	4.37
75%	97.28	85.00	166000	1977.00	42.70	6.00
Max	997.87	100.00	6385382	2015.00	1924.50	6.00

Due to the large number climate variables, summary statistics for those variables hasn't been provided in the report. Please refer to the code.

3.2 Exploratory Data Analysis

In our Exploratory Data Analysis, we aimed to find how individual explanatory individuals relate and behave alongside the response variable, Site EUI, as well other explanatory variables. We aim to suggest potential transformations to the features to find behaviors that will improve performances of models.

Fig. 1: Relationship between explanatory variables and response (Site EUI)

Due to high dimensionality, scatterplots for only the important variables selected (discussed in later sections) are displayed above

3.3 Correlation Analysis

In a preliminary attempt of gathering information on covariate relationships with Site EUI, we chose to perform spearman correlation. We find that energy star rating has the largest negative correlation with Site EUI (-0.66), suggesting a relatively strong decreasing relationship with the response. We also note a weak negative relationship of Site EUI with months from January to March and a weak positive relationship with summer months such as May,June, July, etc.

3.3.3 Building Charactersitics

From the histograms of building characteristics, we saw state factor classes were very imbalanced for State 10 with only 15 datapoints, while State 6 has much more than the others with 50840, indicating that we could undersample for this specific state factor, while combining stat 10 with another state factor level. We also see the disparity in frequency of facility types, where "multifamily (un-categorized)" facilities vastly outnumbered all other facility types. We saw that the building classes were majority residential (57), with the others classified as commercial (43).

Building characteristics are of particular interest in context of the data, and so the researchers investigated relationships between Site EUI and building variables. The most noteworthy of relationships was with energy star rating, which was as expected from the Spearman correlation analysis, and found a clear, more linear relationship with the response than almost all other variables.

For elevation we used binning for the classes based on the variables quantiles from summary statistics. We see a notable increase in site EUI initially as buildings add a floors requirement more total energy usage, before this plateaus, which is understandable in context as buildings aren't generally built above a certain height and number of floors.

3.3.4 Investigating Relationships between explanatory variables (Pairwise Correlation)

It is in our interest to explore potential relationships between covariates in the dataset, as these could potentially lead to issues namely multicollinearity. We did this before any transformations and again used spearman coefficient for the same reasons as before. The monthly minimum, maximum, and average temperatures for colder months display relatively strong relationships with climatic variables relating to cooler weather including "days below…", "heating degree days" and "snowfall" covariates. We gather that the strong negative relationship in context shows as temperature increases in these cold months, the amount of snow fall and days below 0F and 20F decrease as expected. The relationship is the same for warmer months, where cooling degrees days increases with for example `august_average_temperature`,

The high correlation between months of the same seasons suggest implementation of additional features for the 4 seasons in the place of separate months for example. the relationships observed between climatic features indicate potential interactions or new replacement covariates in place of these terms

4. Methods

4.0. Pipeline

Below in Fig. 6, we have a Proof-of-Concept pipeline that addresses all of the client's research questions. A breakdown of each of the steps shown in the end-to-end workflow diagram is covered below.

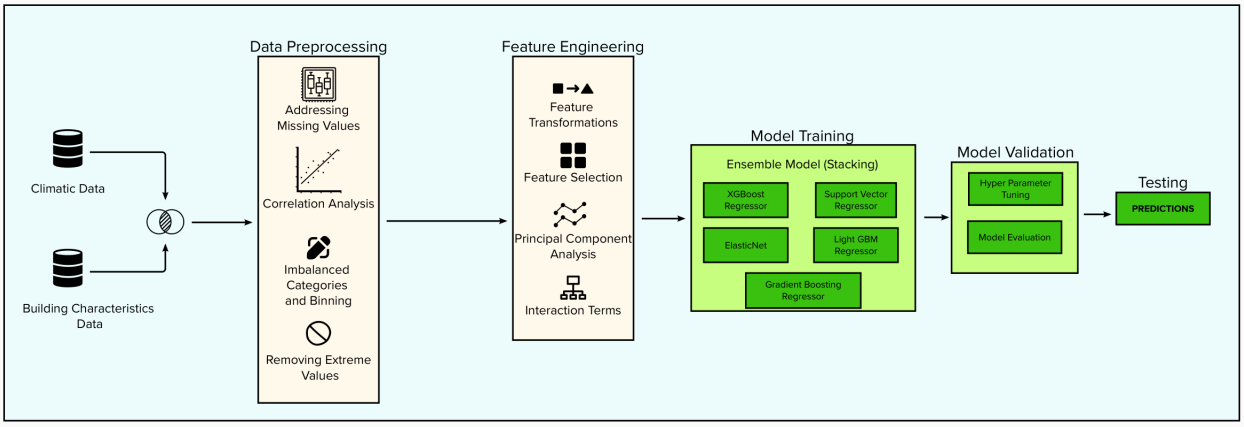

Fig. 6: End-to-End Pipeline

4.1 Data Pre-Processing

As part of the data pre-processing pipeline, we start by dropping duplicate rows, and columns with a high percentage of missing values (over 50). With the remaining missing values, we used the median imputation method (filling missing values of the column with the median). Furthermore, we dropped highly correlated features using pairwise correlation analysis as observed in the EDA. We also dealt with imbalanced categories, generating plots to study the relative frequency distribution of the categorical features. As observed in the EDA we saw imbalanced state factors, particularly with state 6, with the disparity suggesting we drop this altogether. We also noted disparity between size of classes for facility types, so we dealt with this by binning facility types into "Multifamily (un-categorized)" and "other" categories Finally,w e removed rows with extreme Site EUI values using the IQR method and we used an ordinal encoding to encode all categorical columns.

As part of the pre-processing we encoded categorical features using ordinal encoding, where each unique category was assigned an integer value, resulting in a column of integers; 0 to `n_categories-1` (per features)

4.2 Feature Selection

One of the most important step in our pipeline was Feature Selection as it is one of the main objectives. We have used several different methods in order to achieve this objective:

4.2.1. Variable Importance using Random Forest This method is built on a random forest classifier. It ranks features based on their importance measure i.e. Mean Decrease Accuracy (MDA) where higher means more important. MDA measures how much accuracy the model losses by excluding each variable. The more the accuracy degrades, the more important the variable is. From the variable importance plot we see the 3 most importance variables by far are energy star rating floor area and year built. We proposed the notion of potential interactions with these terms later in feature engineering as these were all building characteristic features that could interact in describing the specific buildings. We also note particular importance of Facility Type, Elevation, Days below 20F, February average temperature, January average temperature, as well as building class, relative to the remainder features in the dataset.

4.2.2. Feature Engineering: Interactions and Additional features Based on the variable importance and pairwise correlations we decided on adding interaction terms for future models. We decided on including interactions of;

- floorarea and year built
- floor area and energy star rating
- all 3; floor area, year built, and energy star rating
- floor area and sum of Cooling Heating degree Days

We also implemented new features containing other features that were otherwise correlated with one another, notably monthly temperature variabels. Thus we decided to introduce seasonal terms, where Spring took the average of March April May temperatures, Summer took average of June, July, August, Fall took average of September October November, and Winter took December January and February.

We also introduced terms for "days below.." and "days above.." temperature variables; - Freezing days; total days below 0 10 F - Cold days; total days below 30 and 20 F - Warm days; total days above 80 and 90 F - Hot days; days above 100 and 110 F

Along with a feature covering both snowfall and precipitation: - Snow Rain inches: sum total inches of snowfall and precipitation

In adding these features we dropped the originals contained within these, in turn this would help leave us with more parsimonious models to interpret.

4.3 Feature Transformations

Table 4 shows all the transformations performed in order to achieve the desired results in reducing impacts of skewness of features.

Table 4: Feature Transformations

Variable	*Description*	*Transformation*	*Skewness*
`floor_area`	Floor Area	Sixth Root Transform	Right Skewed
`year_built`	Year Built	Shifted (2020 − year built) Log Transform	Left Skewed
`energy_star_Rating`	Energy Star Rating	Squared Transform	Left Skewed
`ELEVATION`	Building Elevation	Sixth Root Transform	Right Skewed
`avg_temp`	Average Temperature	None	None Obvious
`SpringTemp`	Average Temperatures of March April May	None	None Obvious
`SummerTemp`	Average Temperatures of June July August	None	None Obvious
`FallTemp`	Average Temperatures of September October November	None	None Obvious
`WinterTemp`	Average Temperatures of December January February	None	None Obvious
`floorxBuilt`	Interaction Term Floor Area and Year Built	Sixth Root Transform	Right Skewed
`floor_areaxELEVATION`	Interaction term of building floor area and elevation	Eigth Root Transform	Right Skewed
`floorxEnergy`	Interaction Term Floor Area and Energy Star Rating	Sixth Root Transform	Right Skewed

Variable	*Description*	*Transformation*	*Skewness*
floorxBuiltxEnergy	Interaction Term Floor Area and Year Built and Energy Star Rating	Sixth Root Transform	Non-Normal
floorxHeatCool	Interaction Term Floor Area and Sum of Heating and Cooling Degree Days	Sixth Root Transform	Right Skewed
freezing_days	Total days below 0 and 10 F	Fifth Root Transform	Slight Right Skewed
cold_days	Total days below 20 and 30 F	Square Root Transform	Slight Right Skewed
warm_days	Total days above 80 90 F	Square Root Transform	Left Skewed
hot_days	Total days above 100 110 F	Dropped	Too Few Data

4.4 Correlation Analysis after initial Feature engineering

Having added new features, dropped others, and transformed many, we then checked the correlations for a second time to see if we could deal with any remaining issues. We saw from the triangular correlation heat map that there were still highly correlated features, and dropped these. The dropped features were; 'floorxBuilt', 'floorxBuiltxEnergy', 'floorxHeatCool', 'floorxEnergy', and 'ELEVATION'.

4.5 Feature Selection after Feature Engineering; Boruta package

Furthermore, we proceded with further feature selection using the 'Boruta' package after feature engineering [https://pypi.org/project/Boruta/]. The method uses a "relevant feature selection wrapper algorithm" to find the most "relevant" features using random forest, ranking on imporatnce based on mean absolute standard deviation, allowing us to find and select these. From this final step of feature selection, the most important variables we were left with were:

- building_class
- facility_type
- floor_area
- year_built
- energy_star_rating
- WinterTemp
- cold_days
- warm_days
- floor_areaxenergy_star_rating
- floor_areaxELEVATION
- snow_rain_inches

4.6 Dimensionality Reduction: Principal Component Analysis

Principal Component Analysis is a very useful method based on mathematics and statistics, which reduces the dimensionality of the inputs in the dataset and solves the issue of collinearity (and by extension, multi-collinearity).

Before conducting PCA we performed Min Max scaling because PCA is solved via SVD, and SVD approximates in the sum of squares sense, so if one variable is on a different scale than another it will dominate the PCA.

For a simplified understanding (Kovan, 2021) of how and why we implemented this, we have have a dataset with multiple explanatory variables where the x and y-axes can be thought of as two "components" (or dimensions) either of which the variables can be strongly correlated with. PCA is designed for continuous variables, and it tries to minimize the variance for each component. Instead of only two components, we can have many more.

The variance of each component is taken into account and the aim is to find the number of components that explain 95 of the variance in the dataset (which in our case was 6). There is more complex theory regarding the correlation matrix, eigenvalues and eigenvectors which forms the basis of PCA, however for the purpose of this analysis, a brief explanation felt appropriate.

4.7 Hyperparameter Tuning

For hyperparameter tuning, the reduced and transformed datasets were used. We used random search method with 10-fold repeated cross validation where we defined a search space or parameter space as a bounded domain of hyperparameter values and the random search randomly sample points in that domain to give an optimized value that performs the best out of all combinations. For example for ElasticNet, we are tuning it based on 24000 combinations of parameters. We have the definition provided here for all the hyperparameters we used, if you need more information on this, you can check the sklearn documentation on each of these models or each out to me and i can direct you to the right resources.

4.8. Model Training and Validation

A good variety of models were implemented as a part of our analysis for extensive results. A set of models were trained to capture the effect of interaction terms on the predictive performance of the model. These models were cross-validated (10-fold CV, Repeated 3 times) and their hyperparameters were fine tuned using Random Search. Please refer to Table 5 and Table 6 for results that denote the predictive performance of the models.

4.8.1. ElasticNet

The Elastic Net is a regularized regression method that linearly combines the penalties of the Lasso and Ridge regression methods. In Elastic Net regression the alpha term is a ratio of penalties $\lambda_1 : \lambda_2$ where λ_1 is alpha value of lasso and λ_2 is for ridge. When setting the ratio $= 0$ it acts as a Ridge regression, and when the ratio $= 1$ it acts as a Lasso regression. Any value between 0 and 1 is a combination of Ridge and Lasso regression Tables 5 and 6 show the results for this method.

4.8.2. Support Vector Machines

To add further complexities to the previous models, we trained Support Vector Machines that expanded our feature space using different kernels. Radial kernel support vector machine is a good approach when the data is not linearly separable. We have a radial kernel to compare performance of models without the complexities relating to Linearity respectively. In the radial kernel, only the neighboring behaviour of data is taken into account which means only those data points influence the modelling compared to the Linear SVM whose performance is similar to a Linear model. The idea behind generating non-linear decision boundaries is that we need to do some nonlinear transformations on the features X_i which transforms them into a higher dimensional space.

4.8.3. Tree Models

We used three tree-based models, XGBoost, Gradient Boosting and Light Gradient Boosting to improve the

performance of our model. Boosting slowly learns unique patterns in the data by sequentially combining individual, shallow trees. LightGBM is a much more optimized version of the gradient boosting algorithm. It produces much more complex trees by following leaf wise split approach rather than a level-wise approach which is the main factor in achieving higher accuracy. LightGBM uses a novel technique of Gradient-based One-Side Sampling (GOSS) to filter out the data instances for finding a split value while XGBoost uses pre-sorted algorithm & Histogram-based algorithm for computing the best split. Here instances are observations/samples. From Table 5 and Table 6, we see that out of all tree-based models, XGBoost performs the best, with the lowest RMSE of 23.8852 and better indicators (Interval Score, Average Length and Coverage of Prediction Intervals).

4.8.4. Ensemble Models

We trained an Ensemble model with interaction terms which combines all the models above to produce improved results. Combining these models will generally tend to produce more accurate predictions than a single model. We used stacking which looks at the weighted RMSE of all the individual models to give a better prediction because it is designed to ensemble a diverse group of strong learners.

5. Model Results

5.1. In-Sample Model Metrics Comparison

Based on the boxplot for R squared values attained from the models we have mentioned, as we expected the Stack Ensemble model has the highest, followed by XGboost, but the most noteworthy takeaway is that we see ElasticNet by far performs the worst possibility to the linear nature of the method. As for the RMSE, we see that Ensemble yet again performs the best with the lowest values, and once again followed by XGBoost. Unsurprisingly ElasticNet performs the worst out of the models. Another metric we considered was how long it took for the models to run and produce output from the training data; "fit time". we see that the Stack ensemble takes generally 800 minutes to 1000, while similarly performing model XGBoost only takes a fraction of the time, and so based on practicality of the method in turning around results, XGBoost would be more preferrable of all methods based on its relatively strong performance and faster run time.

Fig. 17: Comparing Insample Model Perfomance Metrics for Different Models

Fig. 17: Comparing Prediction Performance for Different Models Without and With Interaction Terms

5.2. Prediction Interval Comparison

Table 3: "50 % Prediction Intervals"

Model	Level	Avg Length	Interval Score	Coverage
LGBM	0.5	23.675	58.053	0.494
SVR	0.5	23.266	57.622	0.502
GBR	0.5	24.705	58.114	0.506
Elnet	0.5	29.156	65.155	0.501
XGB	0.5	23.856	57.693	0.505
Stack	0.5	24.338	57.656	0.505

Table 4: "80 % Prediction Intervals"

Model	Level	Avg Length	Interval Score	Coverage
LGBM	0.8	54.604	89.740	0.798
SVR	0.8	53.120	90.000	0.793
GBR	0.8	54.755	89.175	0.795
Elnet	0.8	64.465	96.675	0.799
XGB	0.8	54.165	89.306	0.799
Stack	0.8	54.448	88.648	0.797

Based on the 50 Prediction Intervals provided, the researchers believe that the best model is SVR, as it has the lowest Interval Score and Average Length, while having a coverage of just over 50.

For the 80 Prediction Intervals, the researchers found it more difficult to choose the best model; while XGBoost has a coverage rate closer to 80, Stack Ensemble provides a lower Interval Score. The researchers took another factor that is more practical into account; time. While it took a few minutes to fit the XGBoost model, it took just over 5 hours to run Stack Ensemble, and therefore XGBoost is more practical as a model than Stack Ensemble. Thus based on the in-sample metrics we examined and the prediction intervals, we would prefer XGBoost out of all the methods. Although this is subjective to an extent, it is safe to say yhat ElasticNet, linear method, performed the worst, indicating the usefullness of tree methods for this dataset.

6. Limitations

- After feature selection, there is some potential that the best features for each model type were not selected. When selecting the features that would be included in the hyper-parameter tuning for the models we used the results of both Boruta (random forest based selection method) and Forward Stepwise Regression (regression based selection method). The selected features from each both agreed with each other so they seemed reliable, however we did not do an exhaustive search over all variables for each of the models and it is possible there were better combinations.
- Gridcode being selected as an important feature may lead to poor model performance due to lack of data to properly fit especially in the 'with interaction' case. There are 23 separate gridcodes, so there are only 20-40 observations for each gridcode which is not very much (especially for tree based models). Having access to more data could result in much higher performance.
- Each observation in the dataset is an aggregation of climate data collected over the year (i.e. they are the average values collected over the year). This limits the forecasting power of the predictive models that we have fit as we would need to use the forecasted explanatory variables to predict the stream flow, which will most likely lower the performance of the model.

7. Conclusion

After all the model iterations and improvements, we were able to achieve fairly good results with the Ensemble Model taking into account the interaction effect between variables. These results have large implications when it comes to water resource management economically. We have conducted our primary analysis taking into account the spatial data (e.g. gridcodes) which serves as a good MVP to predict the streamflow.

As a side interest and to expand upon the idea of predicting the streamflow solely using climate variables and not any spatial and temporal data, we conducted an analysis and trained models without this data and the predictive performance dropped significantly. Although this addresses the client's first research question of whether one catchment can be used to extrapolate stream flow in another catchment, the limitation we faced was lack of training data. To be able to model such a complex problem using climatic variables, we need more data to train our model which can help with improving the predictive performance of the model.

To address the second research question of whether or not we can detect the unusual streamflow activity accurately, we built a proof of concept pipeline for the outlier detection system. It will take the features from our prediction model as input and label the observations as either anomalous or regular depending on the anomaly scores which are the measures of deviation from normal behavior. We will face the same challenge here - lack of training data. This will lead to an increase in false positives and false negatives in the outlier detection system which can have detrimental consequences. For example, not being able to detect a subtle increase in the streamflow (false negatives) which could lead to irrigation problems and in severe cases even floods. Or getting a huge pool of outlier values (false positives) that will raise false alarms of anomalous behavior more often than desired.

8. Future Research

There are two areas of further research that would help address the limitations in this study; first is further investigating the missing data and second is using scientific relationships between variables to engineer features. Another approach would be to create lag features to further explore the time related features provided like year_factor, year_built and investigating their effect on site EUI

9. References

- Government of Canada / Gouvernement du Canada. (2021, November 25). Government of Canada / gouvernement du Canada. Climate. Retrieved February 5, 2022, from https://climate.weather.gc.ca/

glossary_e.html
- US Department of Commerce, N. O. A. A. (2012, March 8). Snow measurement guidelines. Snow Measurement Guidelines. Retrieved February 5, 2022, from https://www.weather.gov/gsp/snow
- Janssen, J., & Ameli, A. A. (2021). A Hydrologic Functional Approach for Improving Large-Sample Hydrology Performance in Poorly Gauged Regions. Water Resources Research, 57(9), e2021WR030263.
- Statistical interaction: More than the sum of its parts. Statistics Solutions. (2021, June 22). Retrieved February 21, 2022, from https://www.statisticssolutions.com/statistical-interaction-more-than- the-sum-of-its-parts/

Kovan, I. (2021, September 10). Comprehensive guide for principal component analysis. Medium. Retrieved April 3, 2022, from https://towardsdatascience.com/comprehensive-guide-for-principal-component-analysis-7bf2b4a048ae

10. Appendix

0.0 Python / R Libraries Used:

Python Library Used:

- os
- numpy as np
- pandas as pd
- matplotlib.pyplot as plt
- seaborn as sns
- sklearn import preprocessing
- sklearn.ensemble import RandomForestRegressor
- lightgbm import LGBMRegressor
- catboost import CatBoostRegressor
- sklearn.ensemble import StackingRegressor
- sklearn.linear_model import LinearRegression
- xgboost import XGBRegressor
- sklearn.pipeline import Pipeline
- sklearn.model_selection import KFold
- sklearn.svm import SVR
- sklearn.model_selection import cross_val_score
- sklearn.ensemble import GradientBoostingRegressor
- sklearn import model_selection
- sklearn.metrics import mean_squared_error
- re
- sklearn.linear_model import ElasticNet
- pickle
- warnings

3.3.1 Spearman Correlation with Response

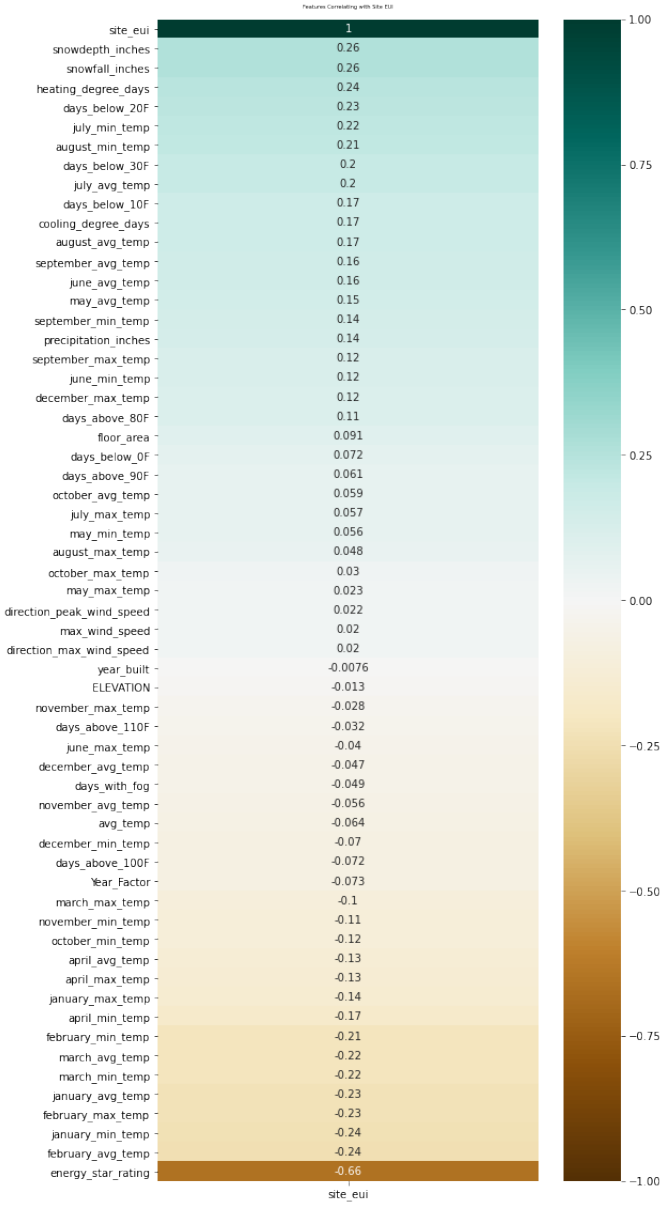

Figure 1: Spearman Correlation with Response

3.3.2 Plots Investigating Relationships with Site EUI

3.3.3 Building Characteristics Histograms

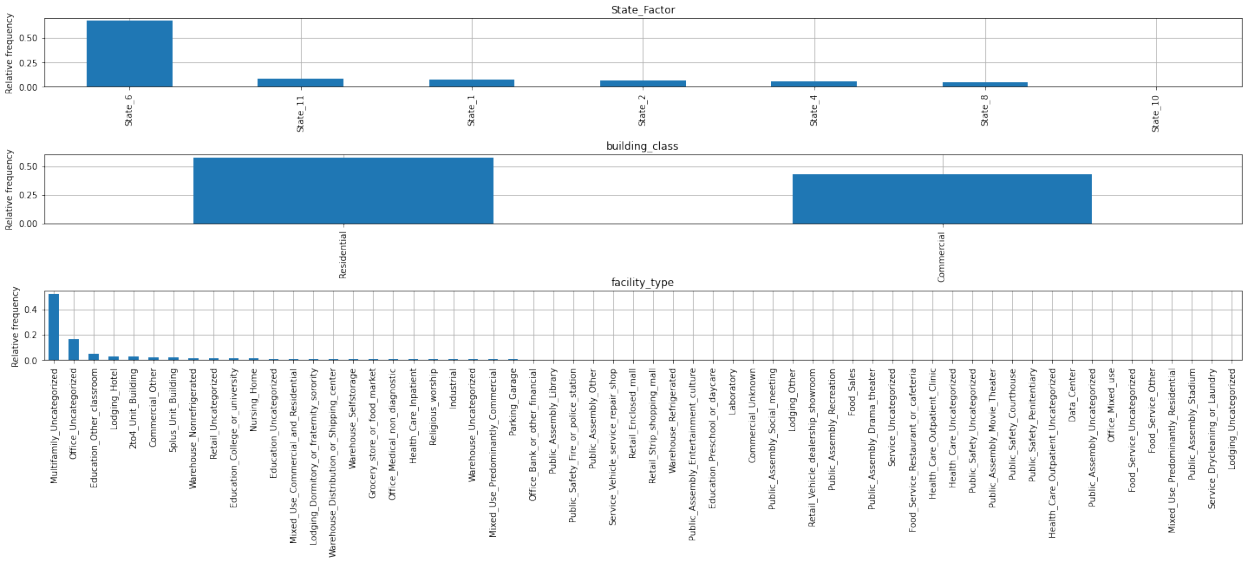

Figure 2: Building Characteristics Histograms

3.3.4 Pairwise Correlation Analysis of Features

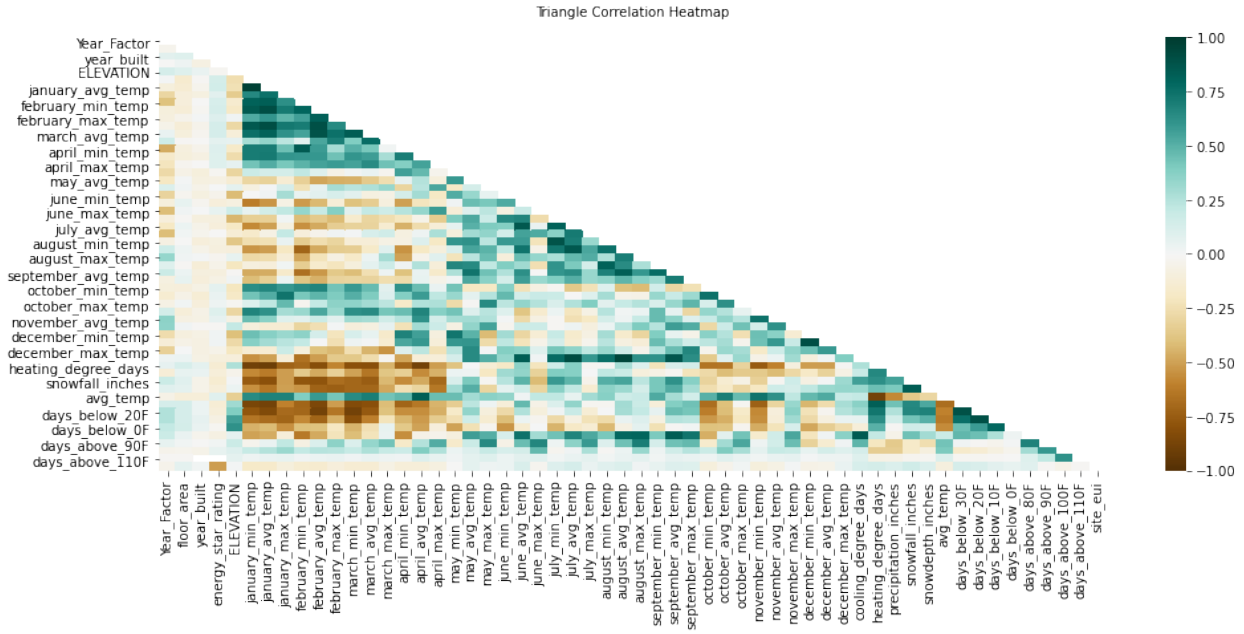

Figure 3: Pairwise Correlation Analysis of Features

4.0.1 Pipeline

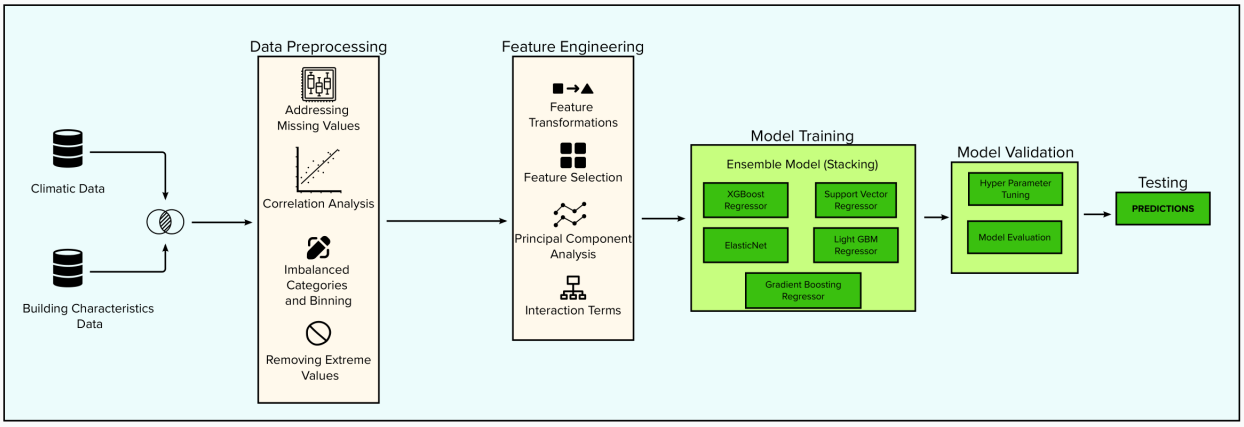

Figure 4: Pipeline of Workflow

4.1 Data Pre processing

4.1 Data Pre processing: Imputation plots

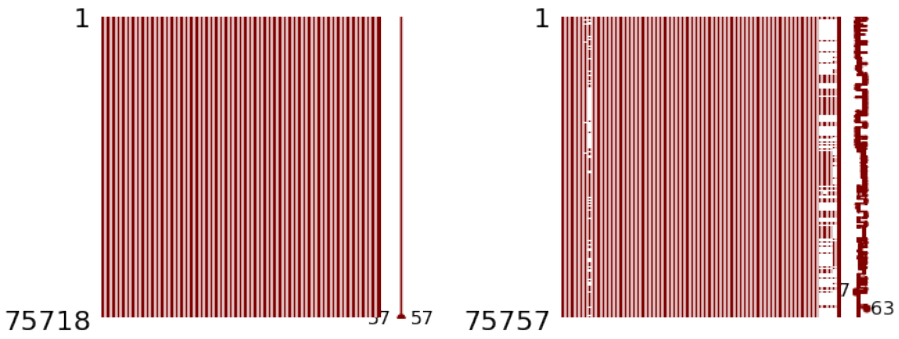

Figure 5: Before and After Median Imputation

4.1 Data Pre processing: Outlier Removal

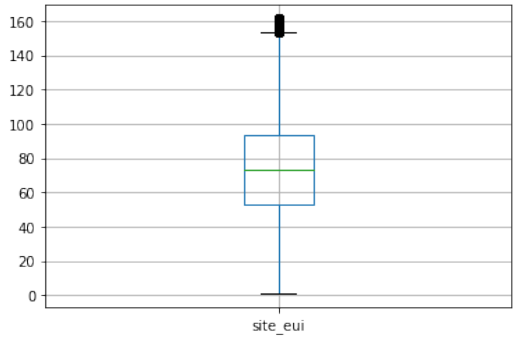

Figure 6: After Extreme Value Removal with IQR Method

4.1.1 Variaple Importance Plots using Random Forest

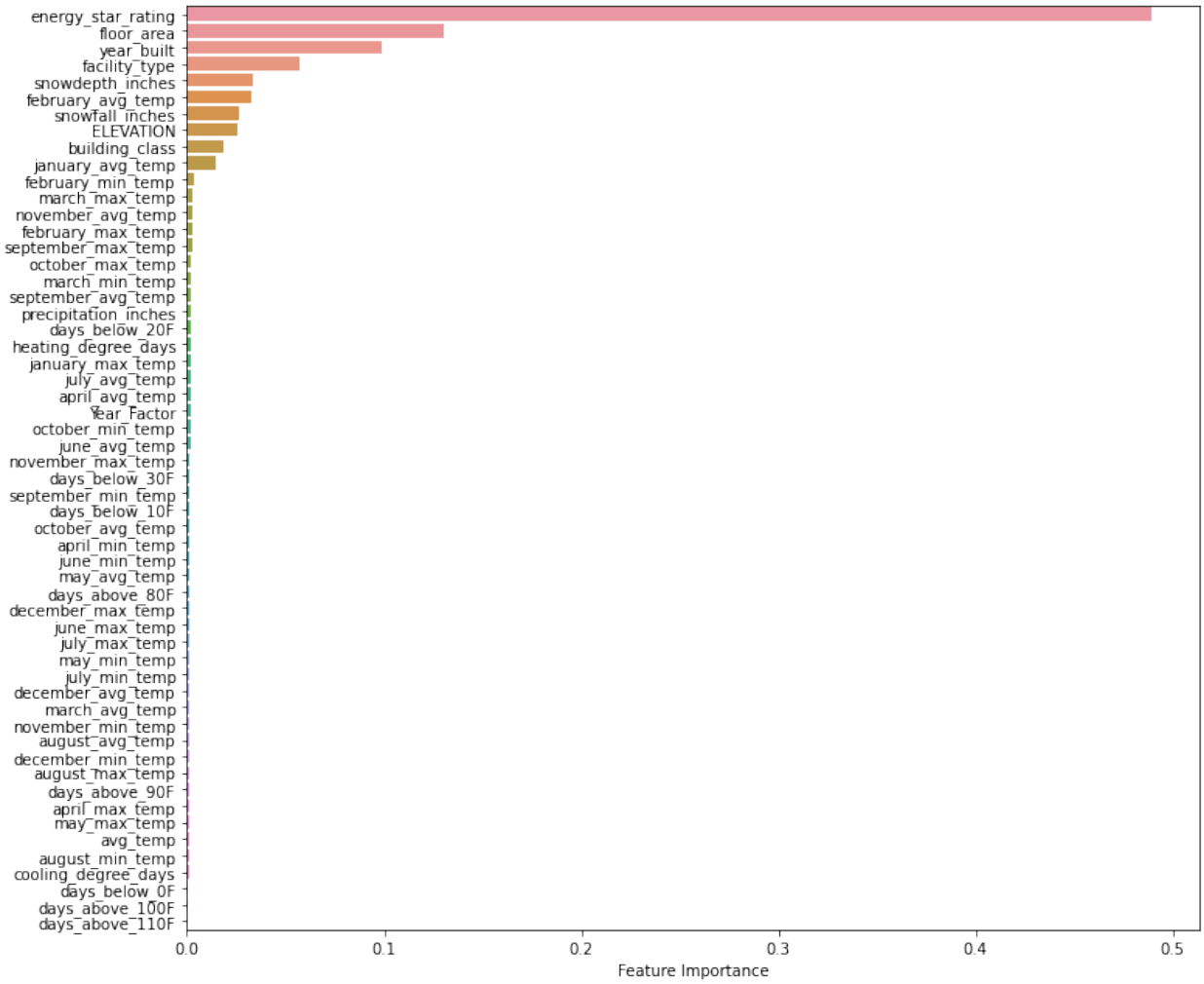

Figure 7: Variaple Importance Plots using Random Forest

4.2.3 Feautre Transformations

4.2.3 Feautre Transformations: Site EUI (no transform needed)

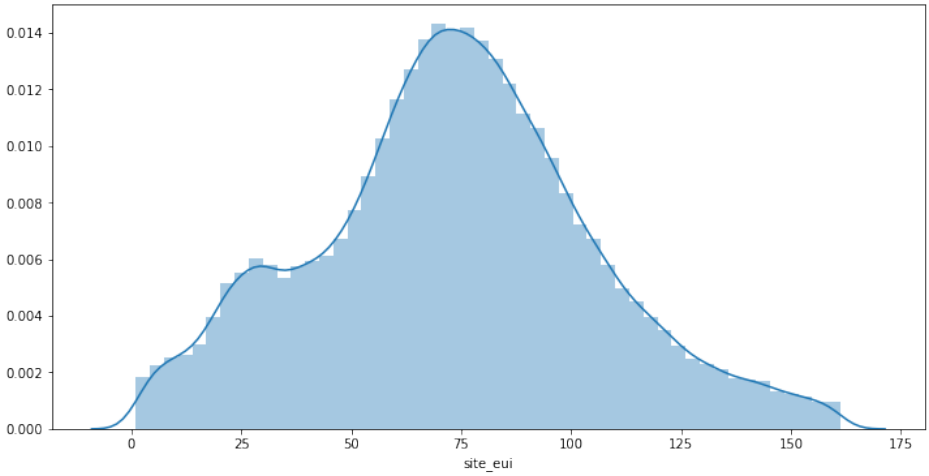

Figure 8: Site EUI (no transform needed)

4.2.3 Feautre Transformations: Floor Area Before and After

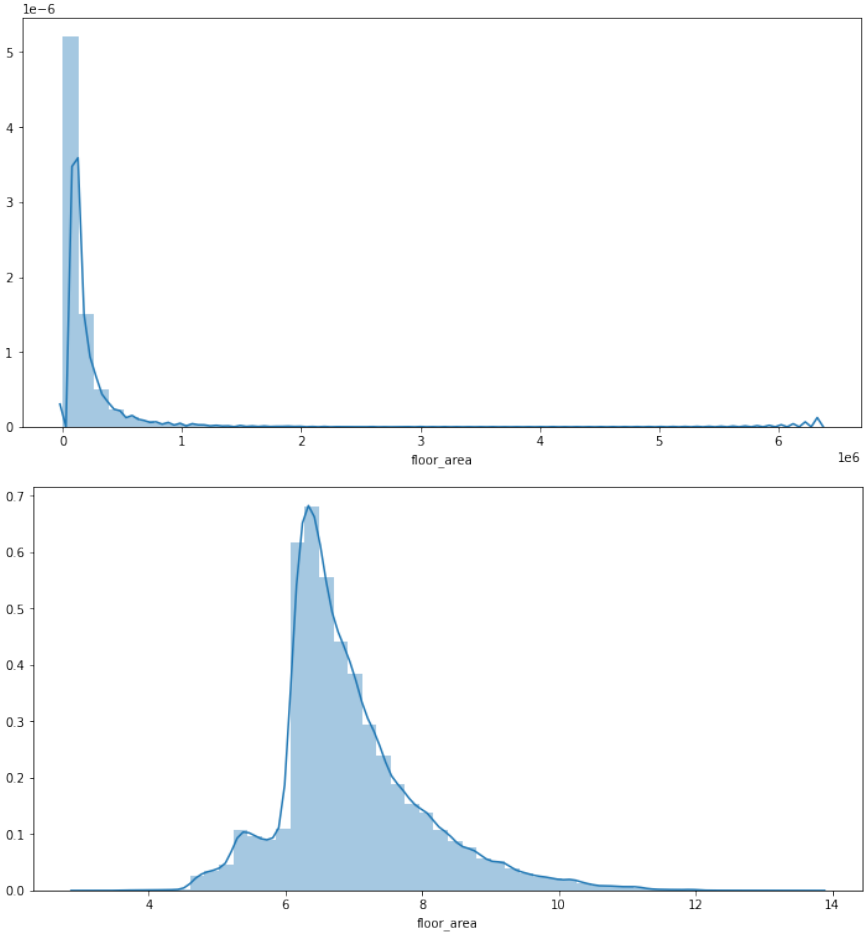

Figure 9: Floor Area Before and After Transform

4.2.3 Feautre Transformations: Year Built Before and After

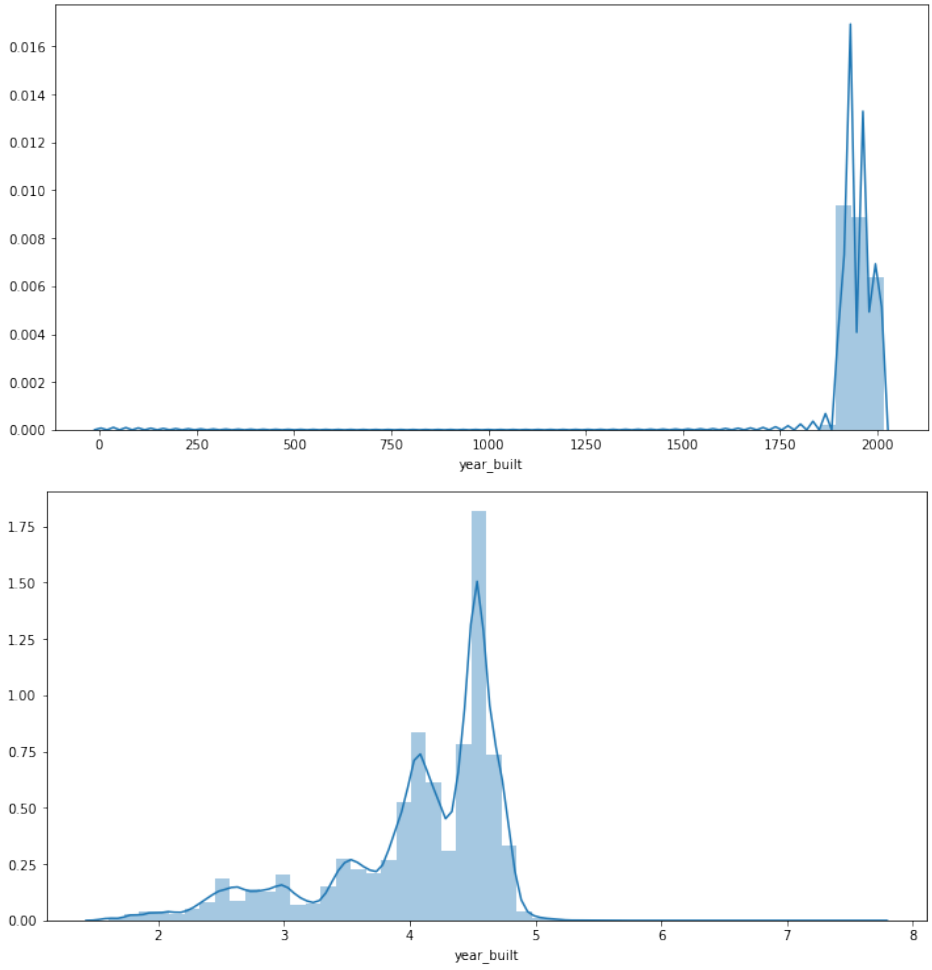

Figure 10: Year Built Before and After Transform

4.2.3 Feautre Transformations: Energy Star Rating Before and After

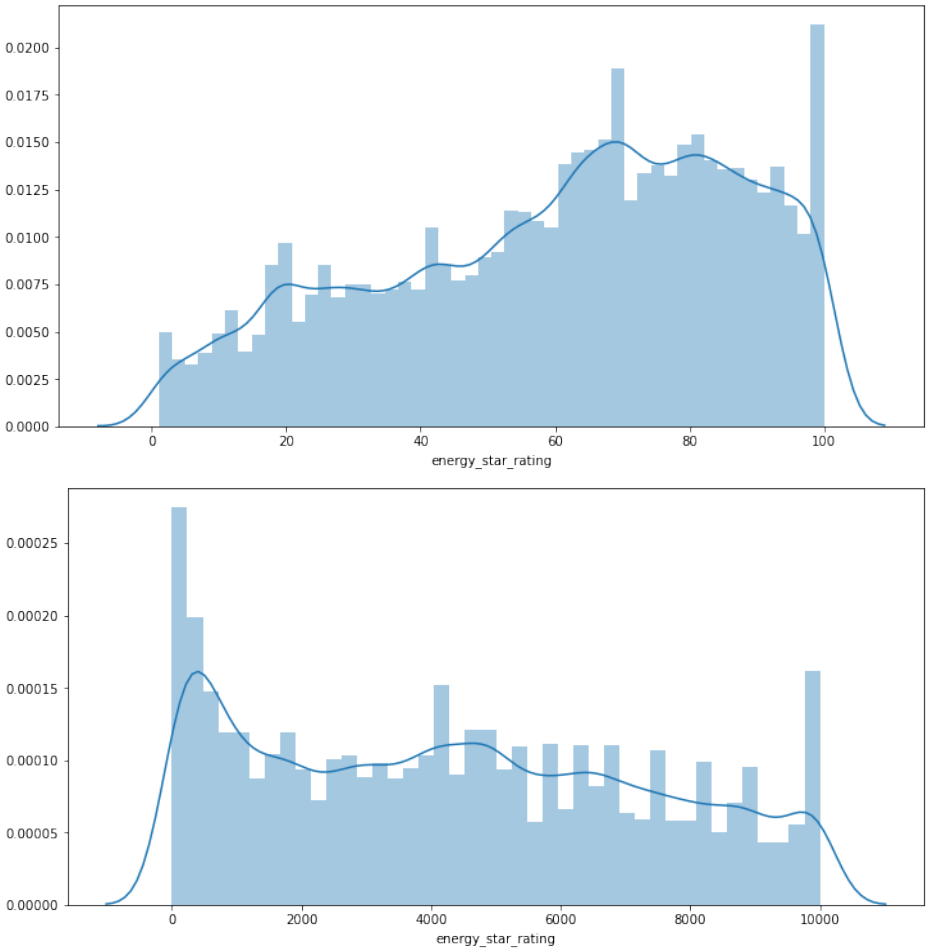

Figure 11: Energy Star Rating Before and After Transform

4.2.3 Feautre Transformations: Elevation Before and After

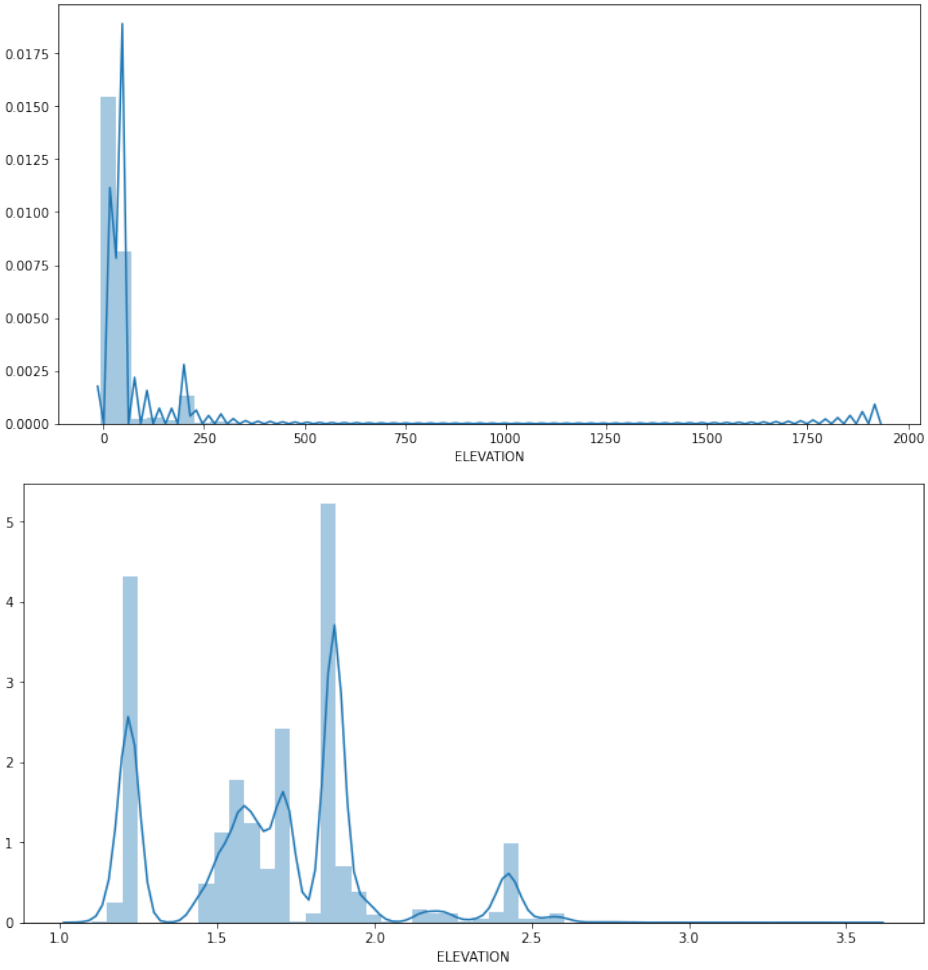

Figure 12: Elevation transform Before and After

4.2.3 Feature Transformations: FloorareaxElevation Before and After

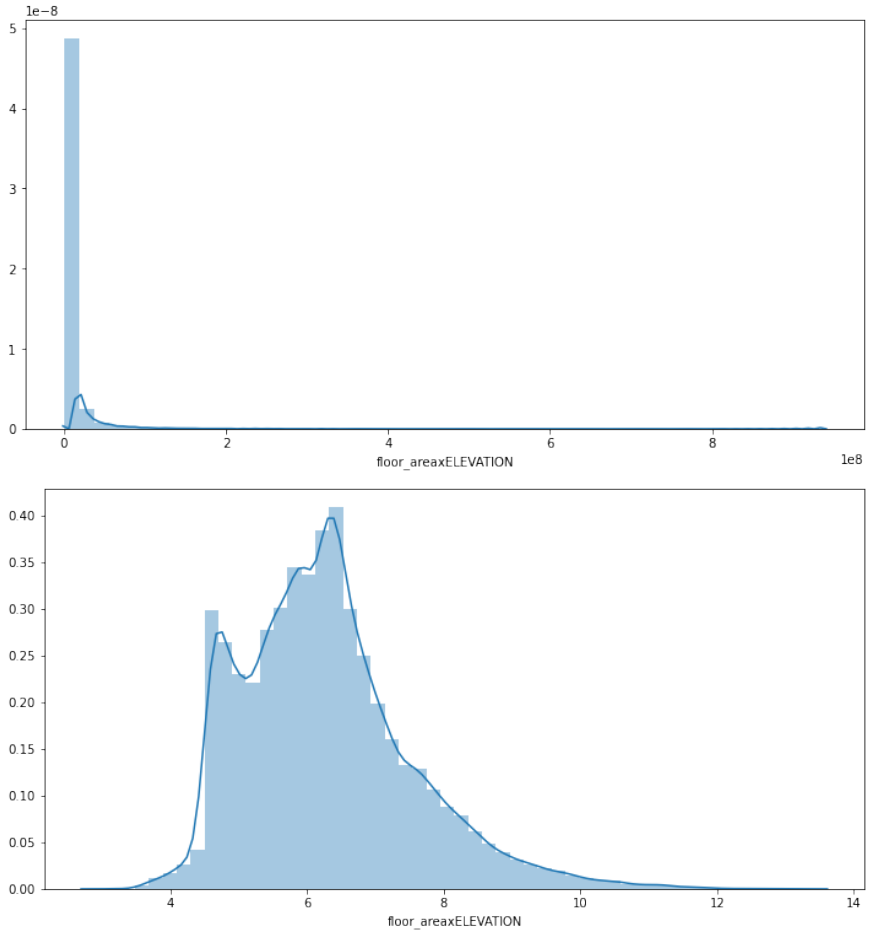

Figure 13: FloorareaxElevation Before and After Transform

4.2.3 Feautre Transformations: FloorxBuilt Before and After

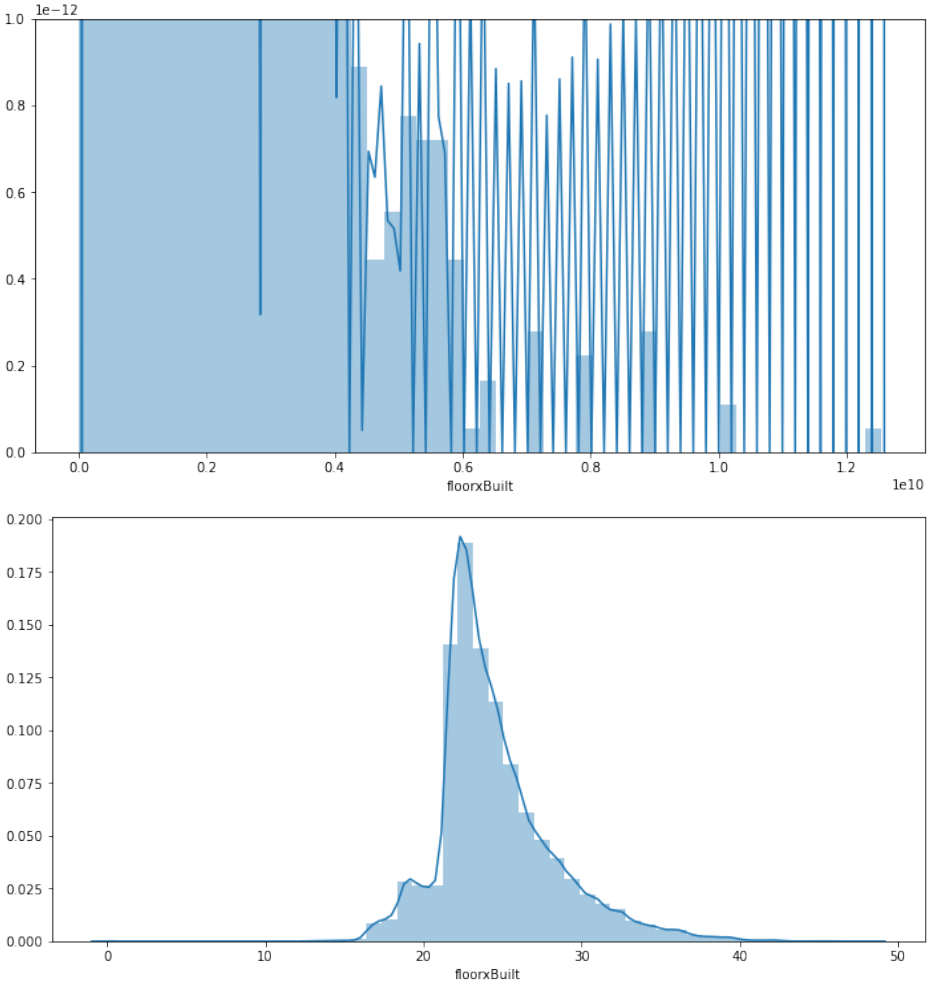

Figure 14: FloorxBuilt Before and After Transform

4.2.3 Feautre Transformations: FloorxEnergy Before and After

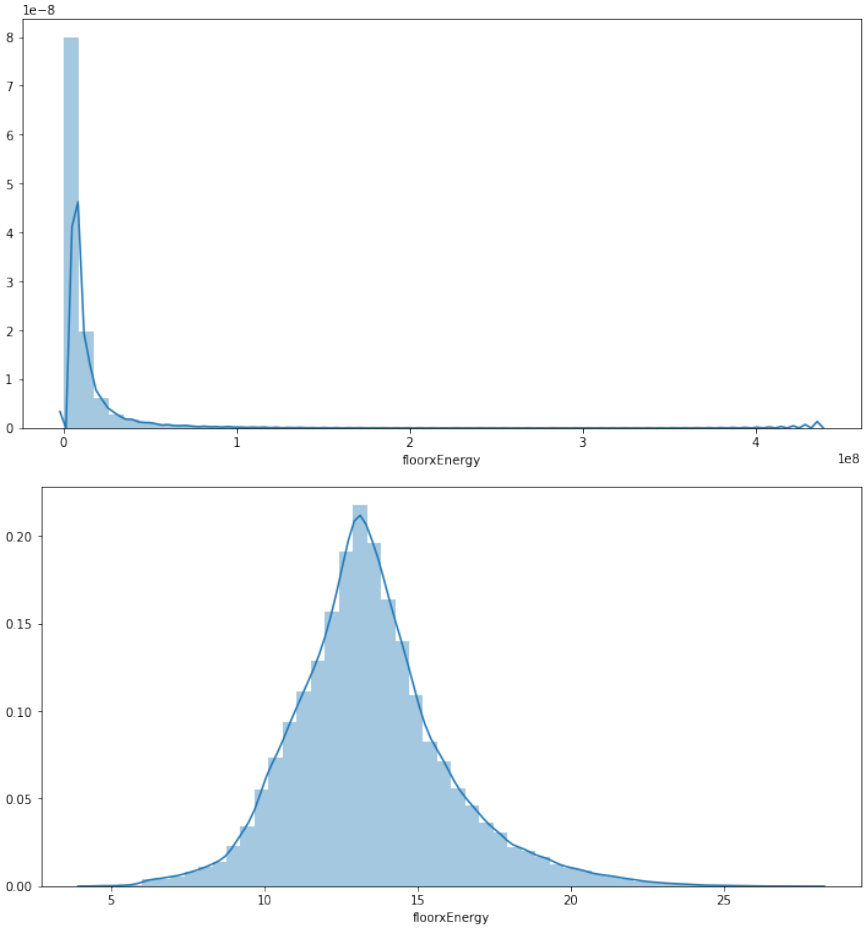

Figure 15: FloorxEnergy Before and After Transform

4.2.3 Feautre Transformations: FloorxBuiltXEnergy Before and After

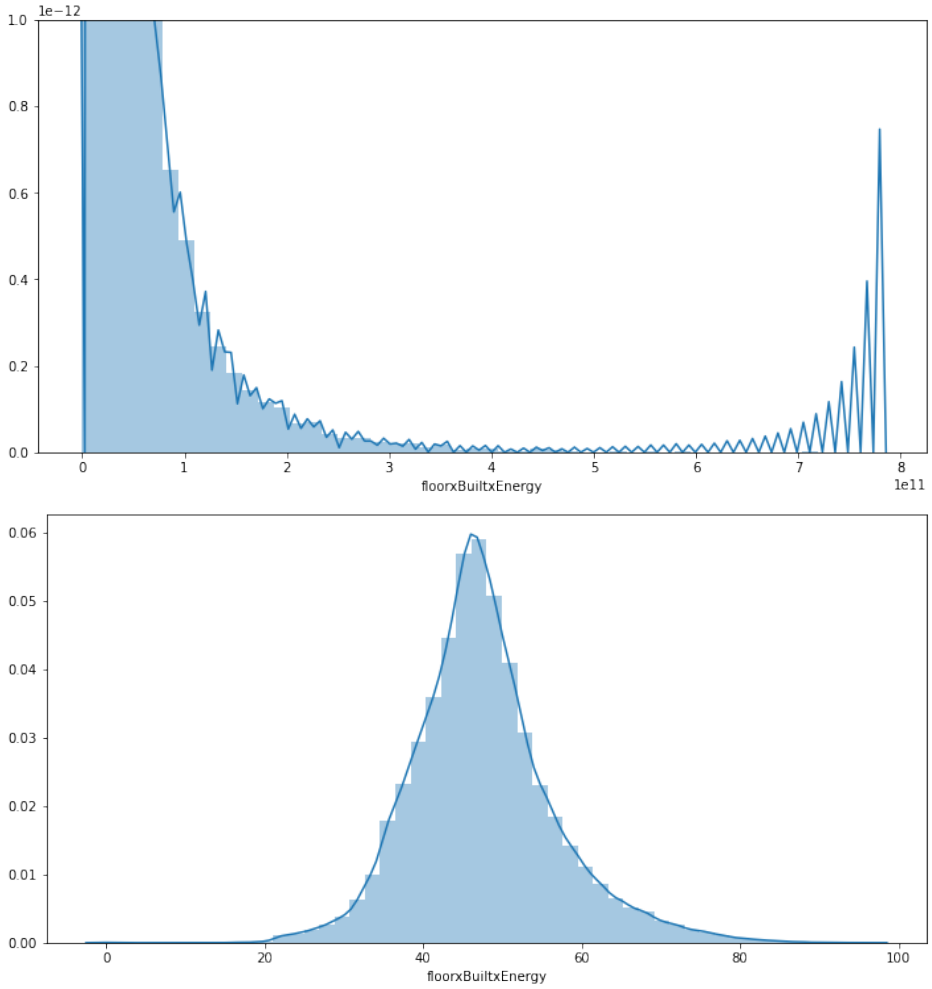

Figure 16: Transforming Interaction of Floor area Year built and Energy star rating, Before and After

4.2.3 Feautre Transformations: FloorxHeatCool Before and After

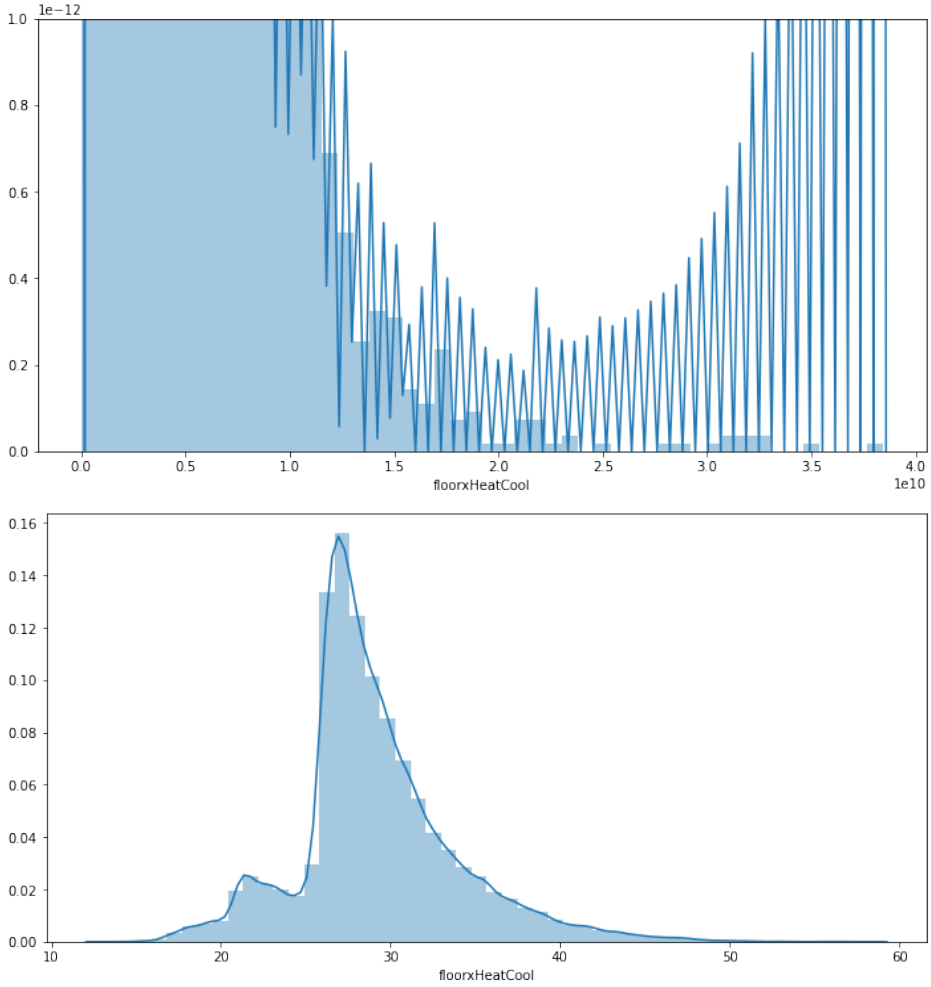

Figure 17: VFloorxHeatCool Before and After Transform

4.2.3 Feautre Transformations: Freezing days Before (no transform needed)

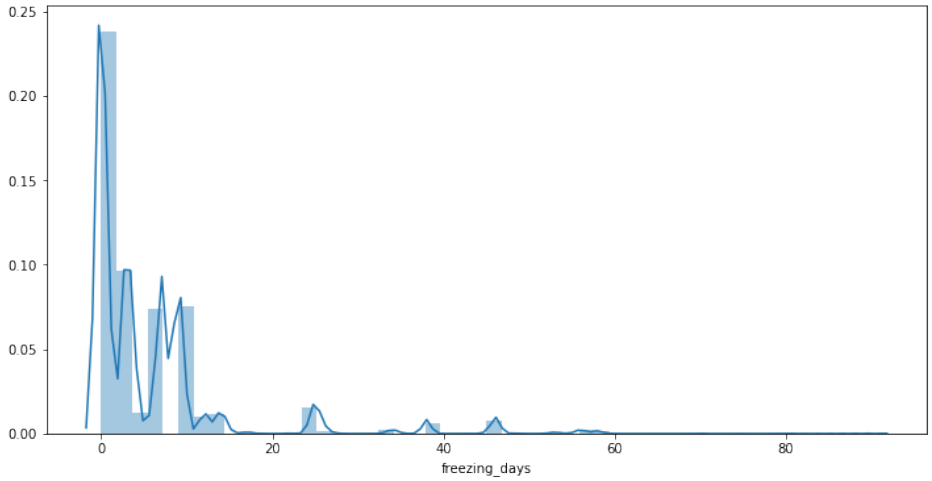

Figure 18: Freezingdays Before (no transform needed)t

4.2.3 Feautre Transformations: Cold days Before and After

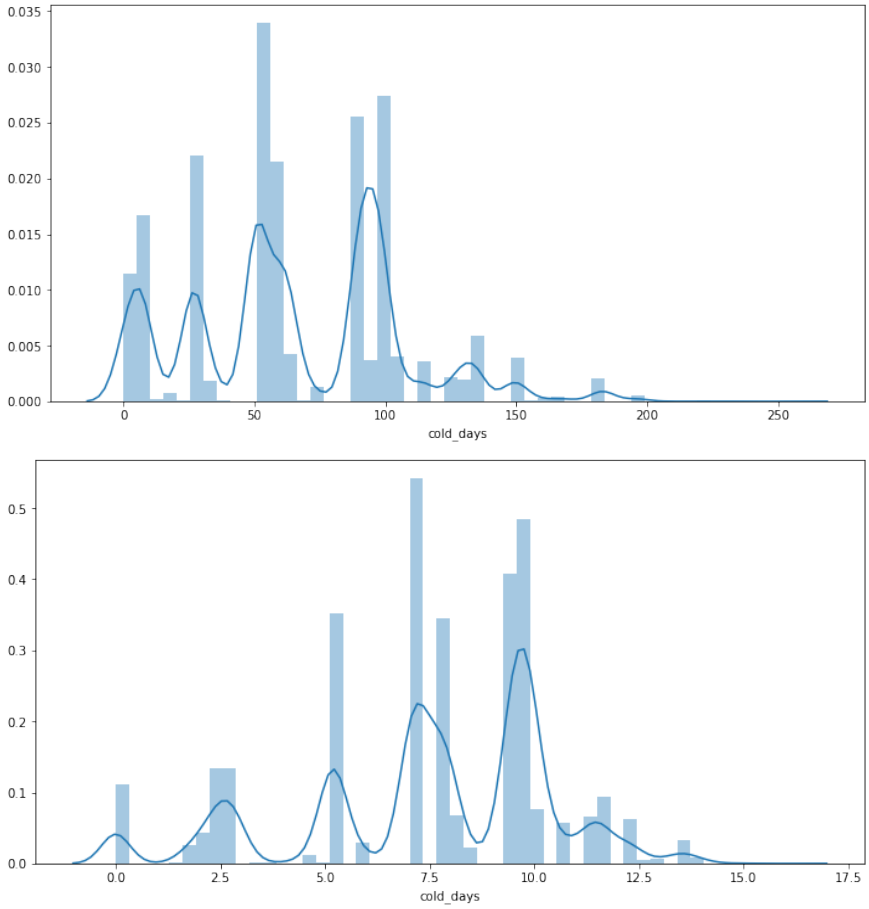

Figure 19: Colddays Before and After Transform

4.2.3 Feautre Transformations: Warm days Before and After

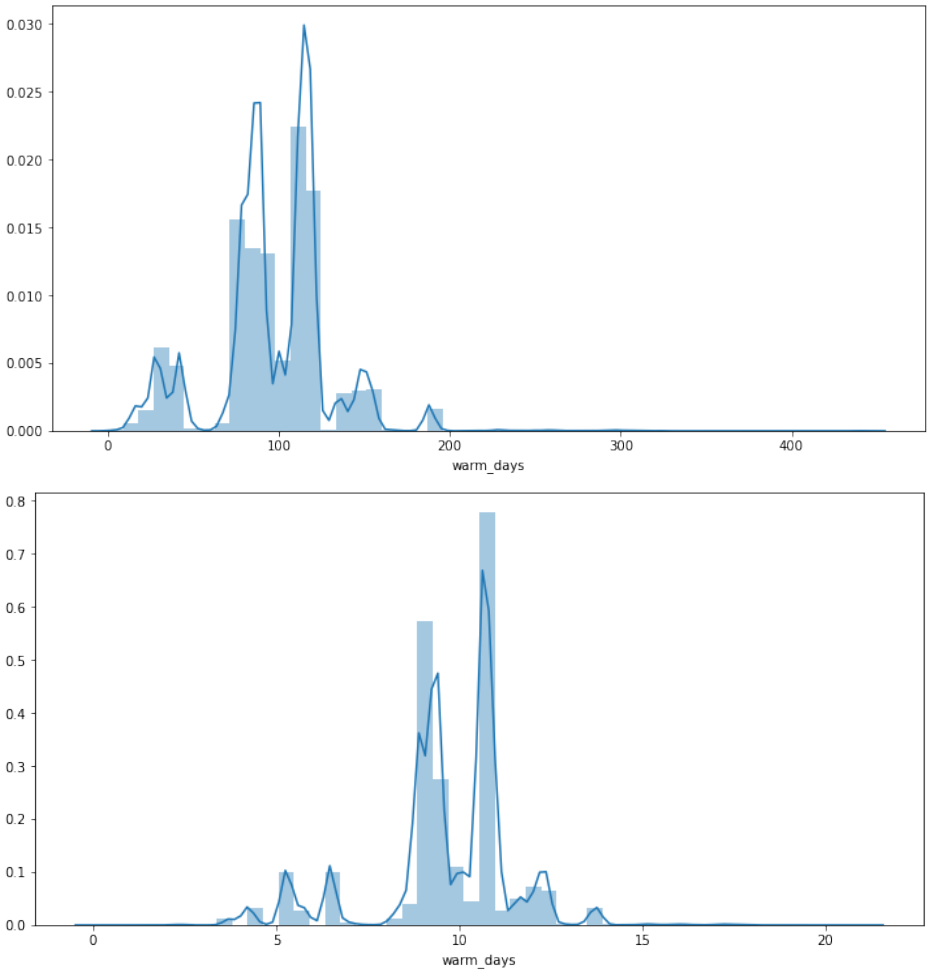

Figure 20: Warmdays Before and After Transform

4.2.3 Feautre Transformations: Hot days Before (no transform needed)

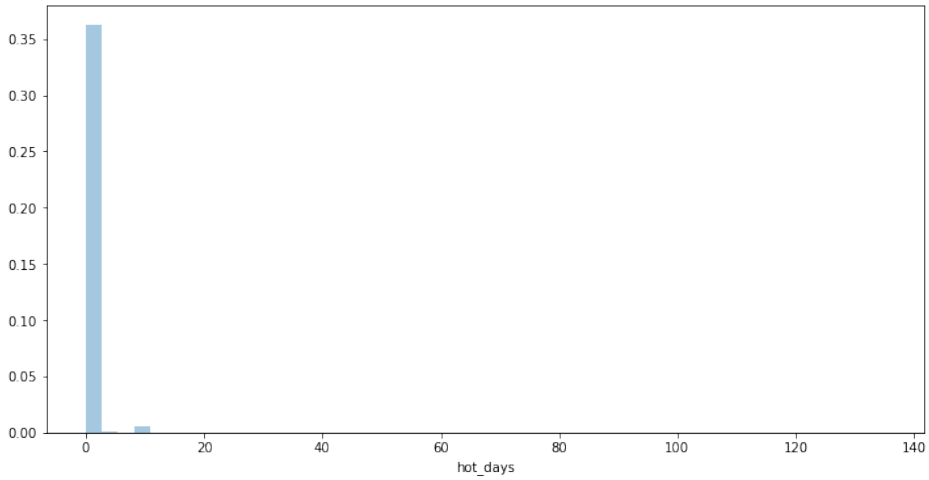

Figure 21: Hotdays Before (no transform needed)

4.2.4: Pairwise Correlation Analysis After intial Feature Engineering

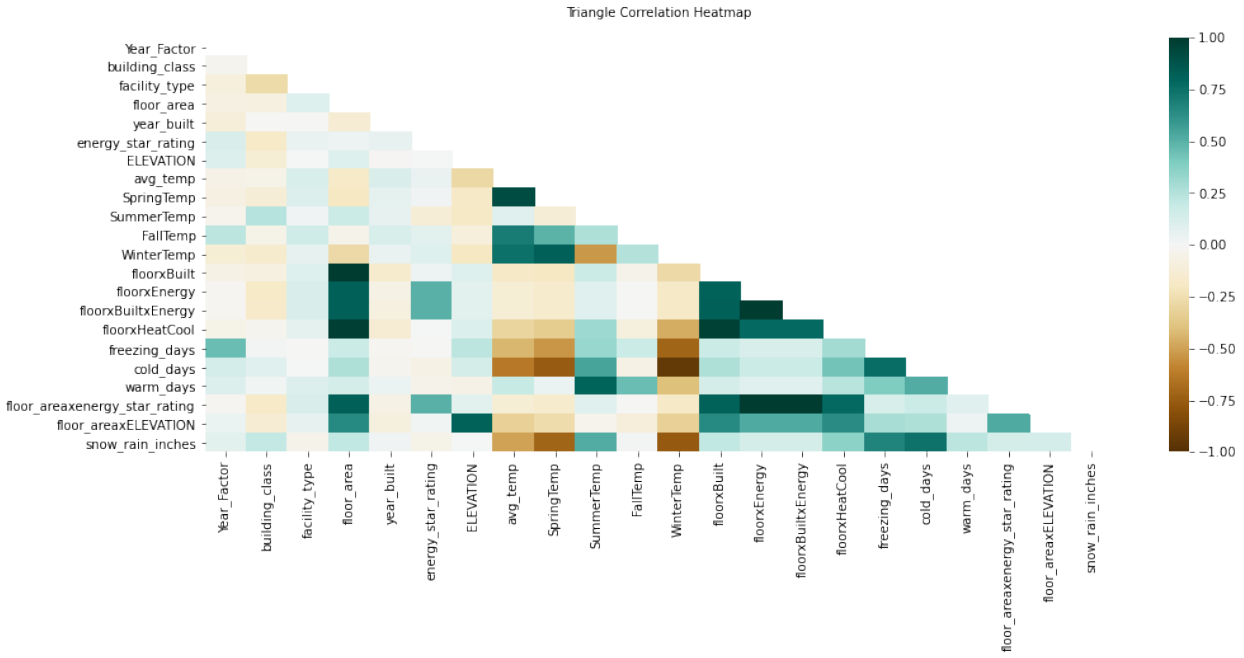

Figure 22: Pairwise Correlation Analysis After intial Feature Engineering

4.2.5: Feature Selectoin After Feature Engineering using Boruta Package

```
Feature: Year_Factor              Rank: 2,  Keep: False
Feature: building_class           Rank: 1,  Keep: True
Feature: facility_type            Rank: 1,  Keep: True
Feature: floor_area               Rank: 1,  Keep: True
Feature: year_built               Rank: 1,  Keep: True
Feature: energy_star_rating       Rank: 1,  Keep: True
Feature: ELEVATION                Rank: 3,  Keep: False
Feature: avg_temp                 Rank: 5,  Keep: False
Feature: SummerTemp               Rank: 4,  Keep: False
Feature: FallTemp                 Rank: 1,  Keep: True
Feature: freezing_days            Rank: 1,  Keep: True
Feature: cold_days                Rank: 1,  Keep: True
Feature: warm_days                Rank: 1,  Keep: True
Feature: snow_rain_inches         Rank: 1,  Keep: True
```

Figure 23: Feature Selection After Feature Engineering using Boruta Package Output

4.2.6 Principal Component Analysis 1/2

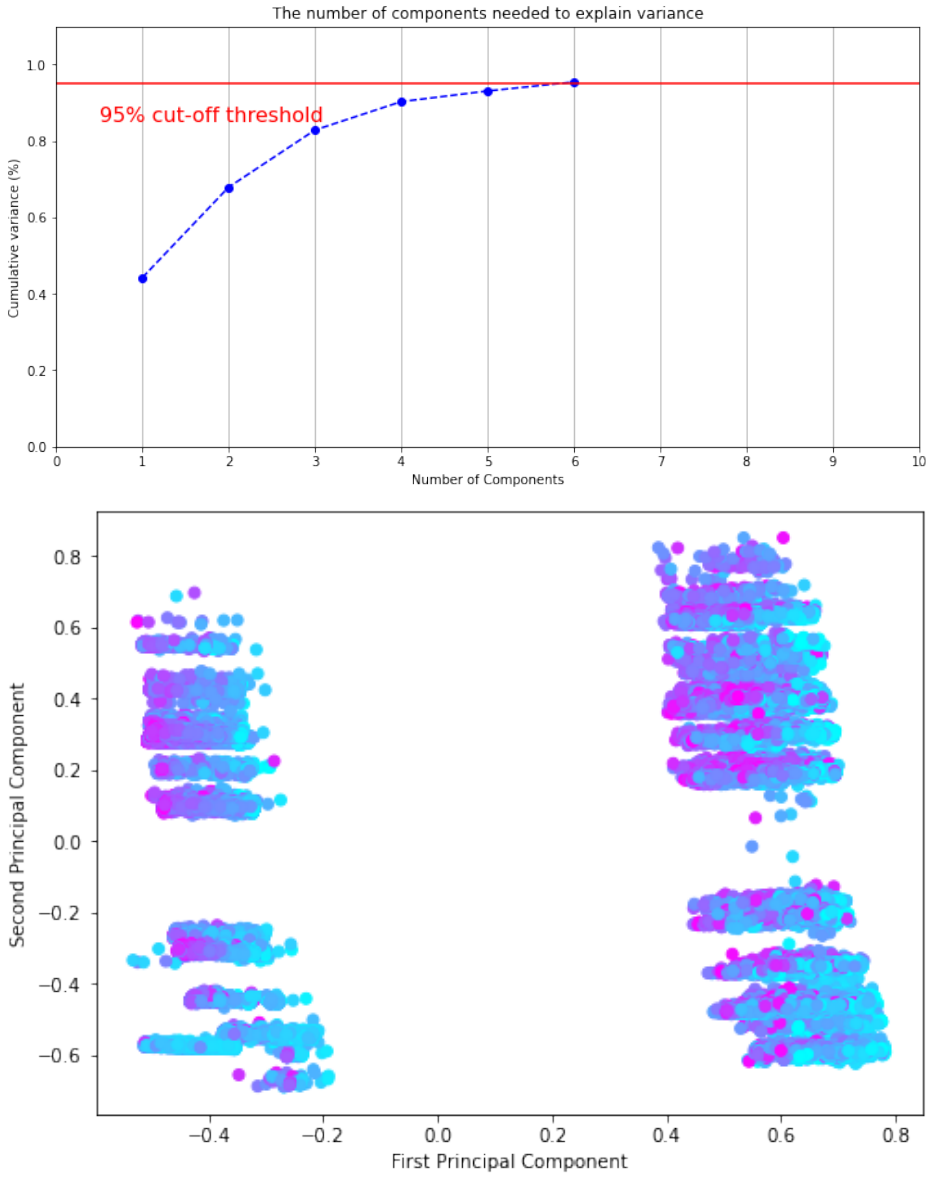

Figure 24: Principal Component Analysis 1/2

4.2.6 Principal Component Analysis 2/2

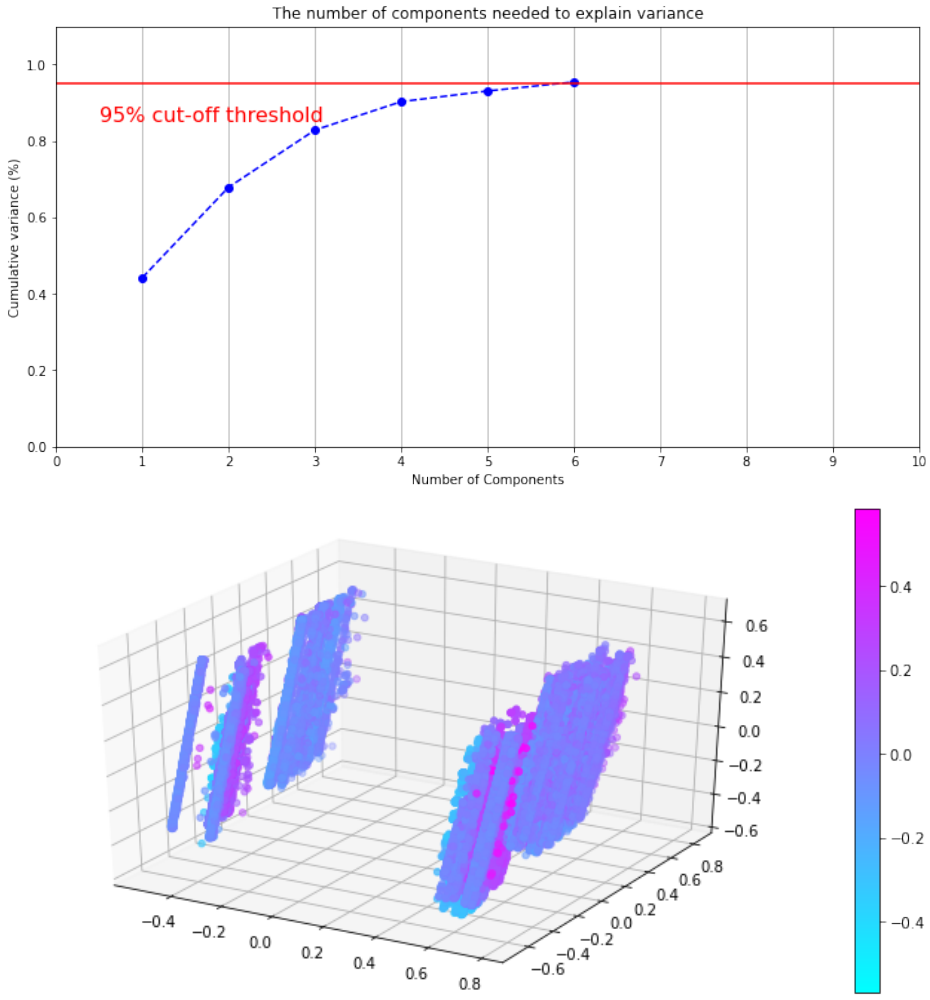

Figure 25: Principal Component Analysis 2/2

4.3.6 Model Results 1/2

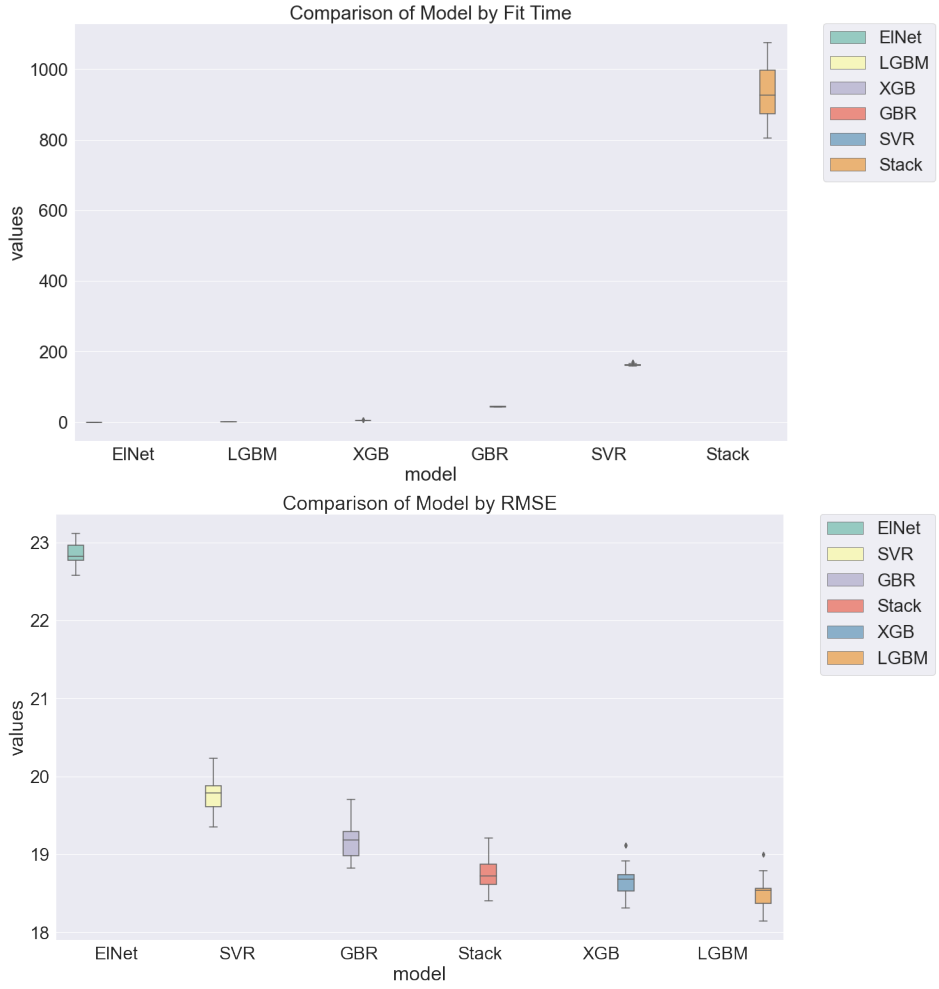

Figure 26: Model Results 1/2

4.3.6 Insample Model Metric Comparison Results 2/2

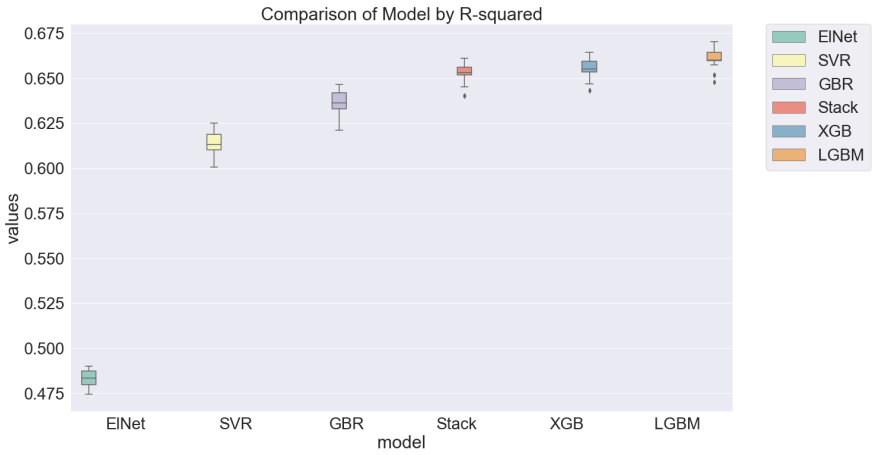

Figure 27: Model Results 2/2

5.0 Model Results: Prediction intervals

Table 3: "50 % Prediction Intervals"

Model	Level	Avg Length	Interval Score	Coverage
LGBM	0.5	23.675	58.053	0.494
SVR	0.5	23.266	57.622	0.502
GBR	0.5	24.705	58.114	0.506
Elnet	0.5	29.156	65.155	0.501
XGB	0.5	23.856	57.693	0.505
Stack	0.5	24.338	57.656	0.505

Table 4: "80 % Prediction Intervals"

Model	Level	Avg Length	Interval Score	Coverage
LGBM	0.8	54.604	89.740	0.798
SVR	0.8	53.120	90.000	0.793
GBR	0.8	54.755	89.175	0.795
Elnet	0.8	64.465	96.675	0.799
XGB	0.8	54.165	89.306	0.799
Stack	0.8	54.448	88.648	0.797