Anticipez les besoins en consommation électrique de bâtiments

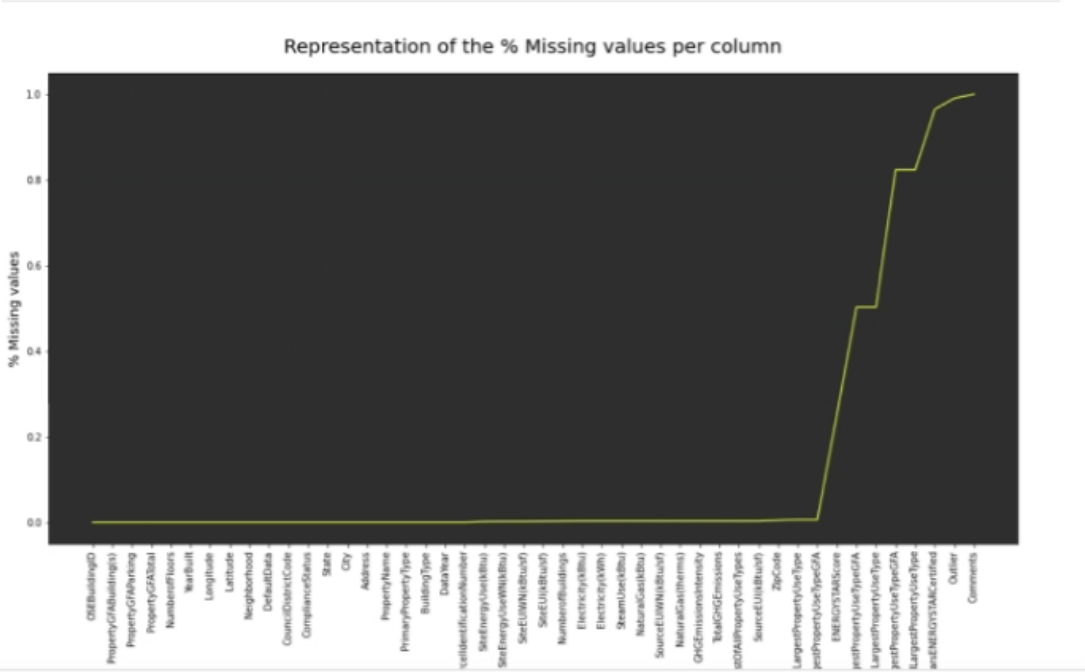
Exploratory Data Analysis (EDA)

1. Imputation Missing Values with Median
2. Observations
3. Removing outliers using IQR
4. Univariate Analysis
5. Correlation Matrix (Pearson)
6. Multivariate Analysis (ANOVA)

Missing Values Observations

We decide to drop the columns that have too many missing values, over **80% missing values** which include:

- *SecondLargestPropertyUseType*
- *SecondLargestPropertyUseTypeGFA*
- *ThirdLargestPropertyUseType*
- *ThirdLargestPropertyUseTypeGFA*
- *YearsENERGYSTARCertified*
- *Comments*
- *Outlier*



Imputation Missing Values

In order to use our model, we have to treat the missing values in the columns that will be relevant for our **feature engineering**.

To do so, we impute our missing values using the **median()** value for our relevant quantitative variables:

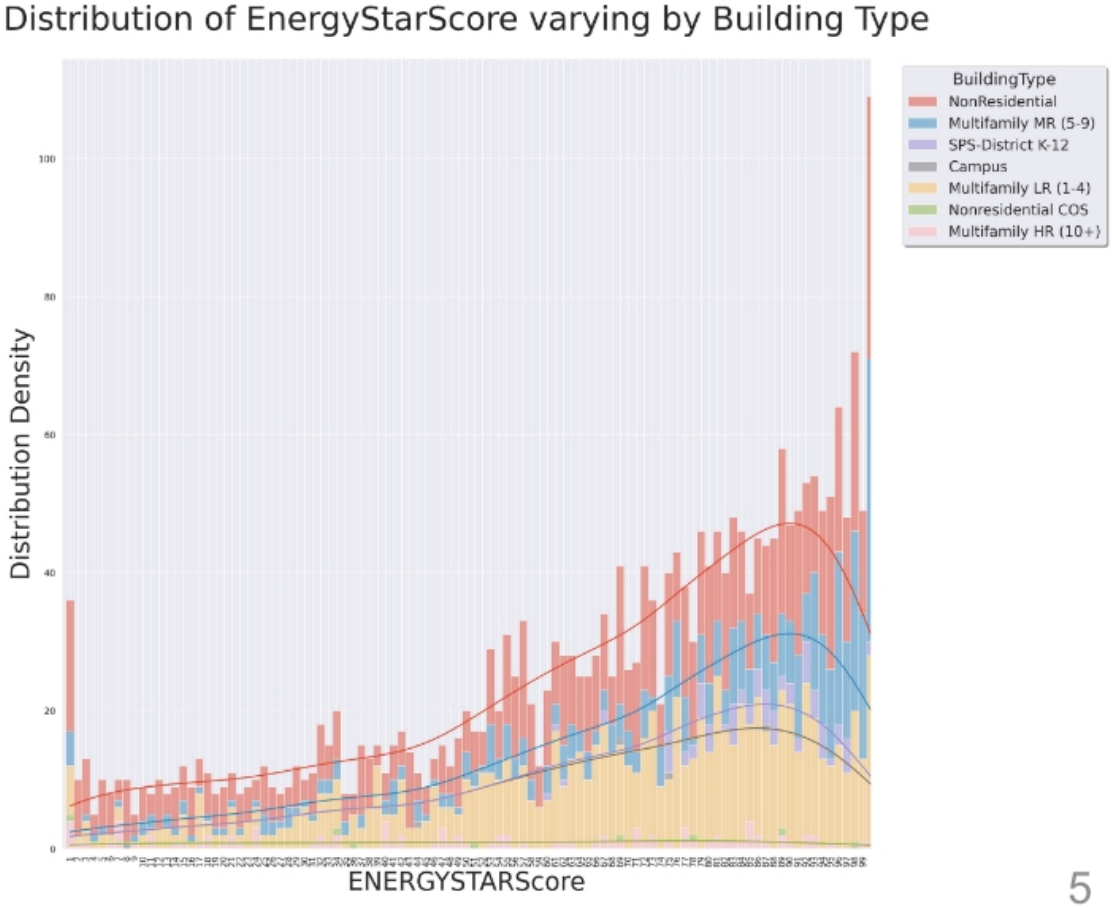
- *PropertyGFATotal*
- *SiteEUI(kBtu/sf)*
- *SourceEUI(kBtu/sf)*
- *SteamUse(kBtu)*
- *Electricity(kBtu)*
- *Electricity(kWh)*
- *TotalGHGEmissions*
- *GHGEmissionsIntensity*
- *SourceEUIWN(kBtu/sf)*
- *SiteEUIWN(kBtu/sf)*
- *SiteEnergyUse(kBtu)*
- *SiteEnergyUseWN(kBtu)*
- *NaturalGas(therms)*
- *NaturalGas(kBtu)*

Observation 1

We notice the distribution of *ENERGYSTARScore* among the *BuildingType*.

We observe that:

- **NonResidential** buildings have the best scores and the highest distribution density
- **Multifamily MR (5-9)** have the second-best scores with the second-highest distribution density
- **Multifamily HR (10+)** have the lowest distribution density and the worst scores

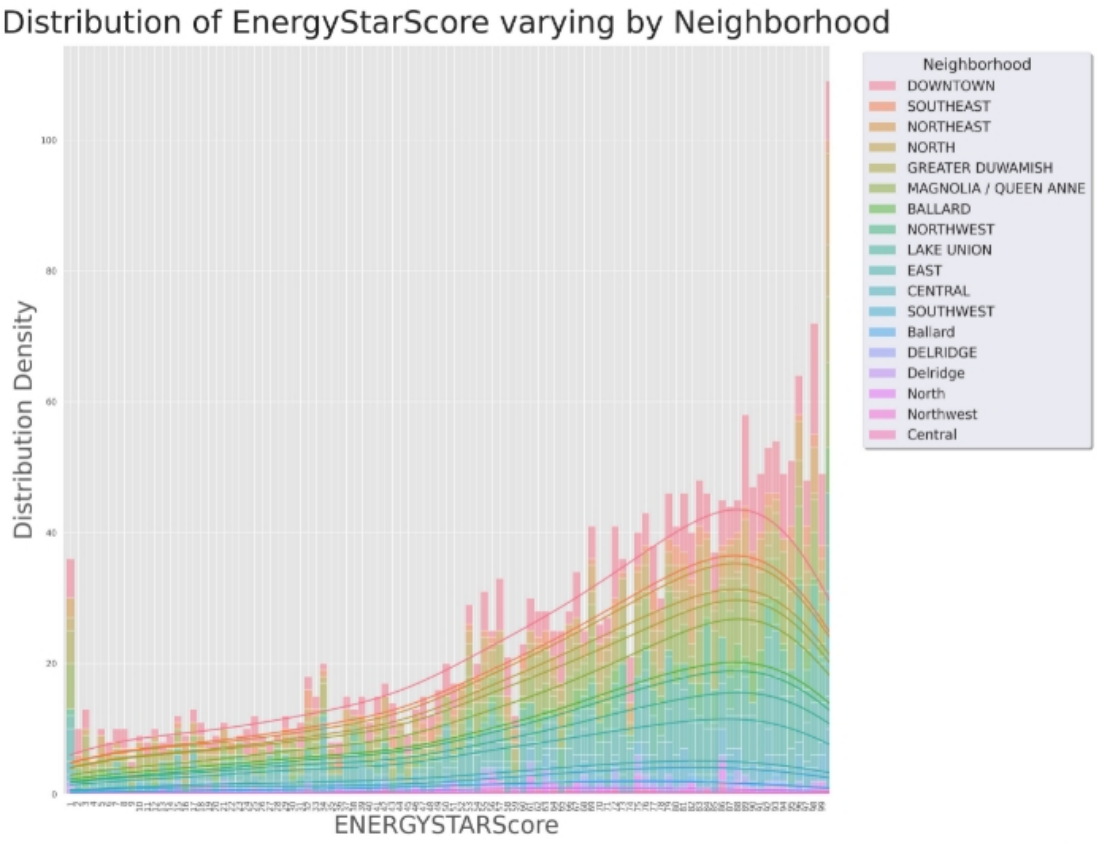


Observation 2

We notice the distribution of *ENERGYSTARScore* among the *Neighborhood*.

We observe that:

- **DOWNTOWN, SOUTHEAST, NORTHEAST** buildings have the best scores and the highest distribution density
- **NORTH, NORTHWEST, CENTRAL** have the lowest distribution density and the worst scores

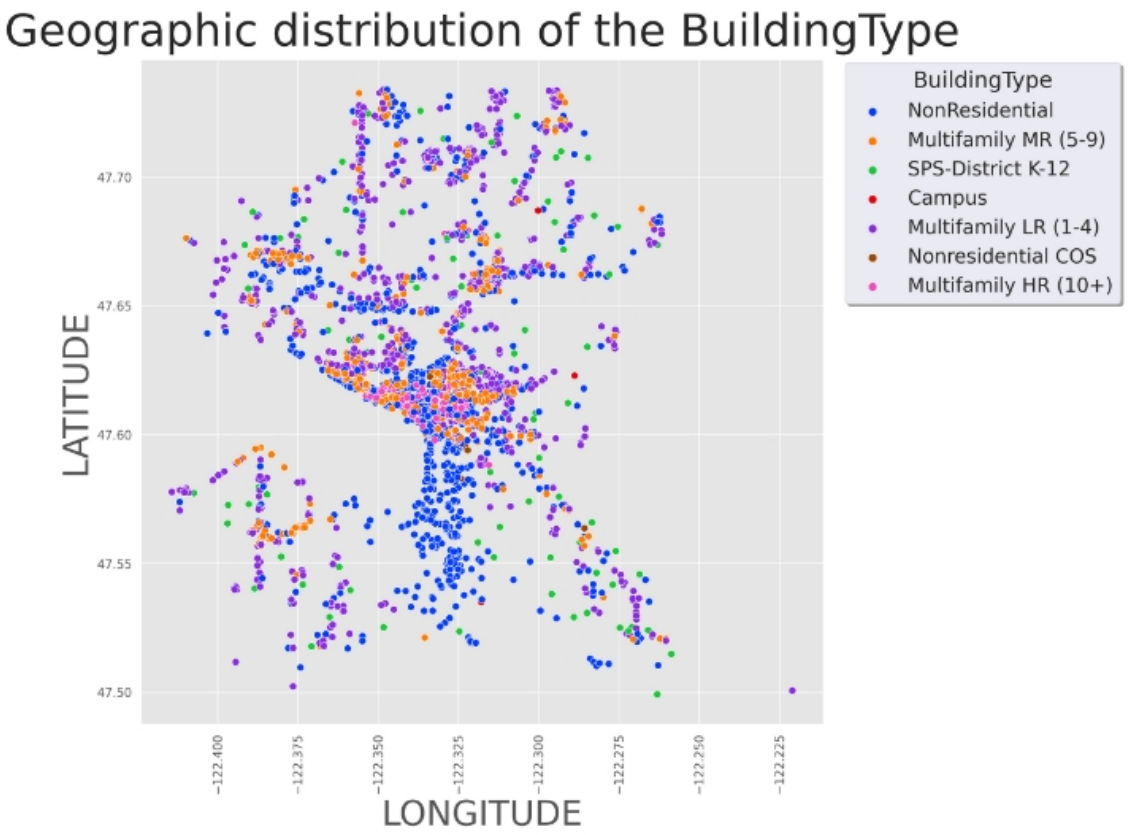


Observation 3

We notice the geographic distribution of the *BuildingType* in Seattle.

We observe that:

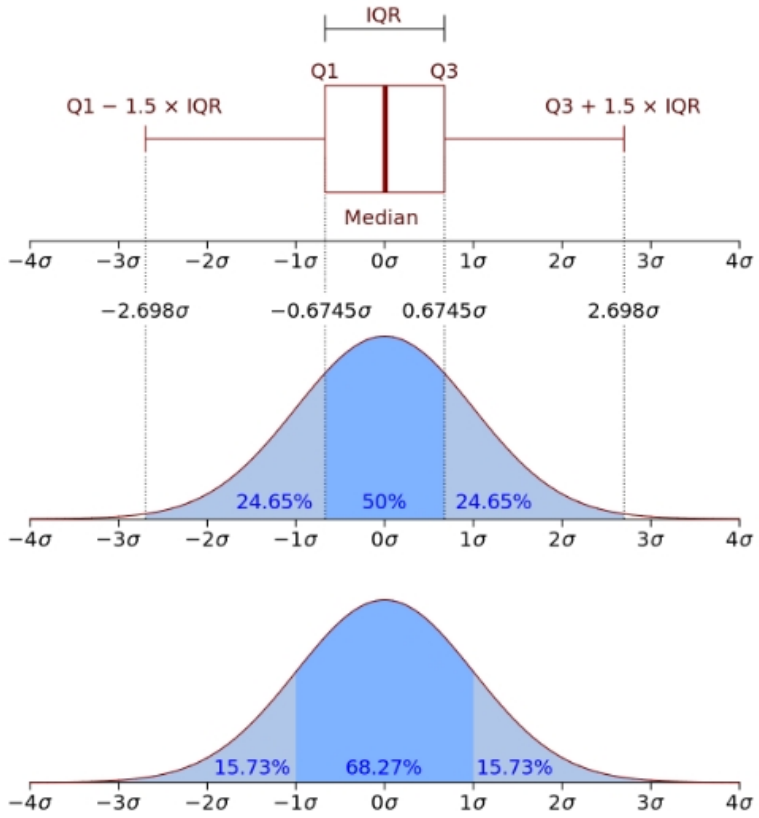
- Most of the buildings are **Non-Residential, Multifamily LR (1-4)**, and **Multifamily MR (5-9).**
- **Non-Residential** buildings are spread in the suburbs of the city, followed by **Multifamily LR (1-4).**
- **Multifamily MR (5-9)** buildings are gathered in the center/downtown of the city.



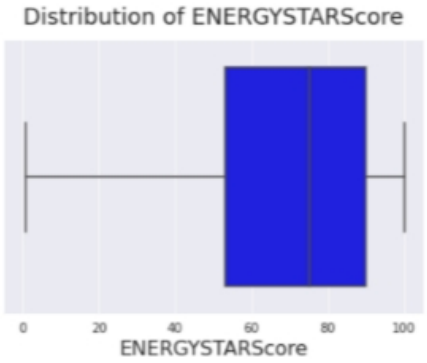
Removing outliers using IQR method

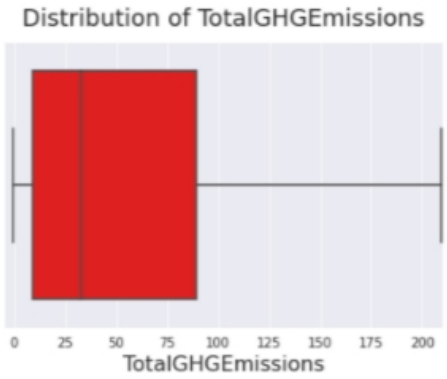
This method allows us to detect outliers in our distribution on quantitative variables. IQR method is based on the **Gaussian distribution** (normal distribution).

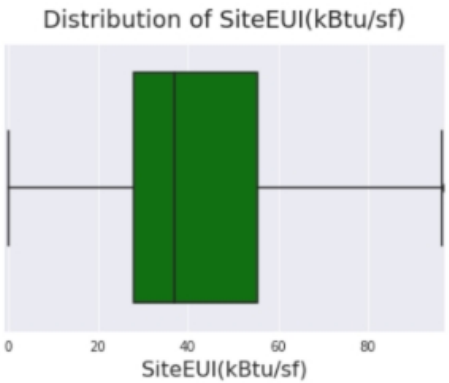
A normal distribution describes the repartition of the data around the mean value and tells us that the whole data is found below three times the standard deviation of the mean value which means that the part of the data above it is considered as outliers: **outliers>3xsigma of mu with sigma the standard deviation and mu the mean**.

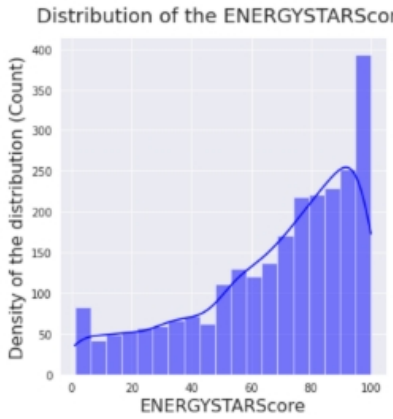


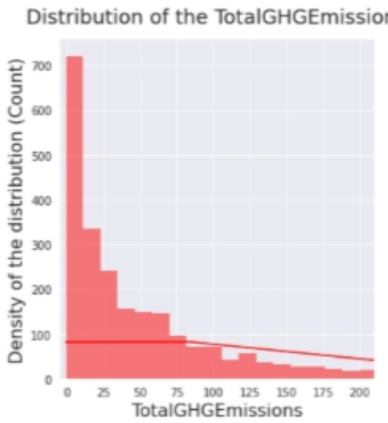
Quantitative variables observations

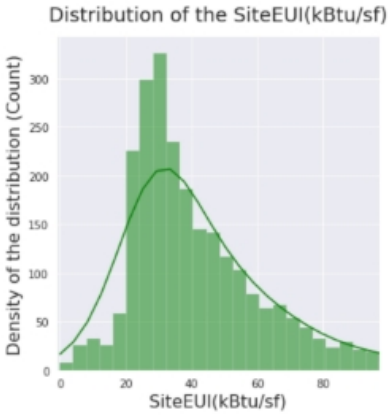




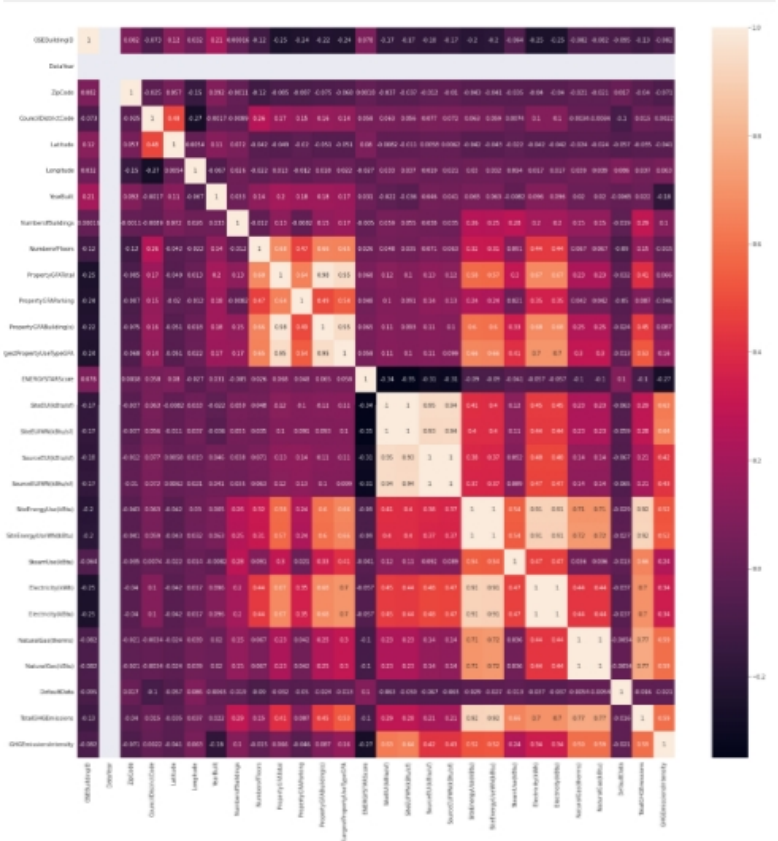








Correlation Matrix (Pearson)

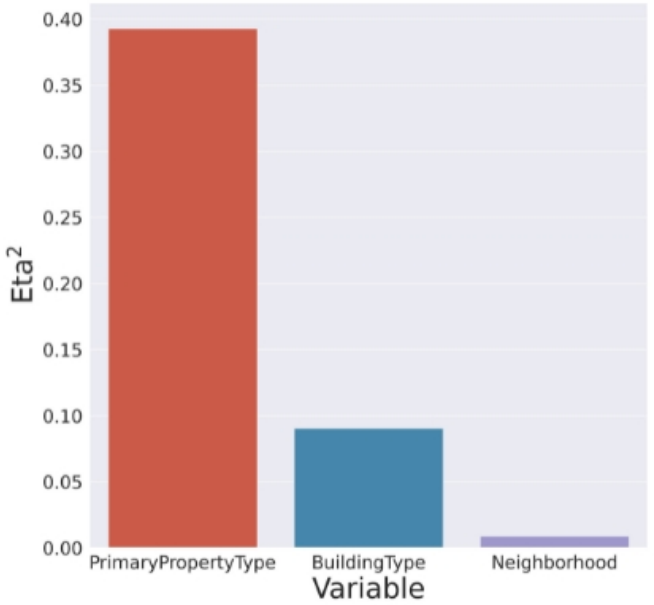


The **correlation matrix** allows us to determine the correlation between the variables and will be helpful for the **features engineering**.

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

Multivariate analysis (ANOVA)

Density of ANOVA results for TotalGHGEmissions versus Categorical Variables



	Variable	Eta^2
0	PrimaryPropertyType	0.39253124973414744
1	BuildingType	0.09035411986523545
2	Neighborhood	0.00869506093222336

ANOVA stands for analysis of variance and, as the name suggests, it helps us understand and compare variances among groups.

We use **Eta squared** to measure the proportion of variance associated with each main effect and interaction effect in an ANOVA model

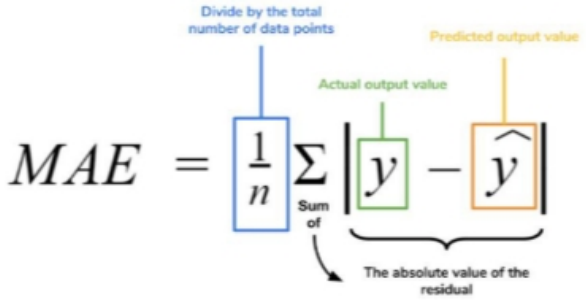
Predictive models

1. Statistical Metrics
2. Features Engineering
3. Cross Validation & Hyperparameters
4. Features Importance
5. Statistical Results
6. Final Model Decision & Features Weight

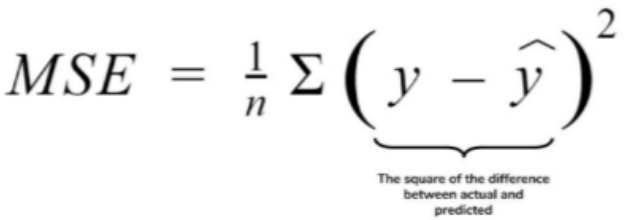
Machine Learning Models

- Linear regression (Least squares)
- Linear regression (Lasso)
- Linear regression (Ridge)
- Linear regression (Elastic net)
- Support Vector Regressor
- Decision Tree
- Random Forest
- Bagging Regressor
- Gradient Boosting
- XGBoost

Statistical metrics #1



Mean Absolute Error represents the average of the absolute difference between the actual and predicted values in the dataset. It measures the average of the residuals in the dataset.



Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

Statistical metrics #2

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Root Mean Squared Error is the **square root of Mean Squared erro**r. It measures the standard deviation of residuals.

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

The C**oefficient of Determination** or **R-squared** represents the proportion of the variance in the dependent variable which is explained by the linear regression model.

Discretization of continuous variables

We use the function **OneHotEncoder()** of scikit-learn library to discretize the categorical variable we've used during our EDA.

At the difference of **LabelEncoder()**, **OneHotEncoder()** does not create a hierarchy in the numerical values. In our case, we don't have a hierarchy in our modalities, so we'll definitely use this class to convert the string into numerical values.

We convert the following variables:

- *BuildingType*
- *PrimaryPropertyType*
- *Neighborhood*

Two notebooks

We create two notebooks in which we'll repeat the same models for **two predictive variables**.
One variable to best predict the **CO2 emissions** and another variable to predict the **total energy consumption**.

We import our features in a dataframe called **X** which will contain all the numerical (discrete) variables that do predict our predictive variable.

As predictive variable to predict the **total CO2 emissions** of property in Seattle, we decide to pick *TotalGHGEmissions* that seems the most representative of our real case study, which means that the lower *TotalGHGEmissions* is, the lower the consumption of the building will be.

To predict the total energy consumption for property in Seattle , we pick **SiteEnergyUse(kBtu).**

Remove correlated variables

Using our correlation matrix, we decide to remove the variables that are too correlated to one another so we avoid future over training of our models.

From this list, we also notice that some variables are highly correlated between them such as:

- *PropertyGFATotal*, *PropertyGFABuilding(s)* and *LargestPropertyUseTypeGFA*
- *SiteEnergyUse(kBtu)* and *SiteEnergyUseWN(kBtu)*
- *Electricity(kWh)* and *Electricity(kBtu)*
- *NaturalGas(therms)* and *NaturalGas(kBtu)*

Cross-validation

To start creating ML models, we gotta do cross-validation of our data which means dividing our dataset into a training set and a testing set.
To do so, usually, we divide our data into a certain number of folds.

In our case, we'll divide our dataset into 5 folds using **KFold(n_splits=5,shuffle=True)**.

We use cross-validation for mainly two purposes:

- Defining our **hyperparameters** to train and test each ML model
- Create **different folds** with two specific purposes: training and test sets

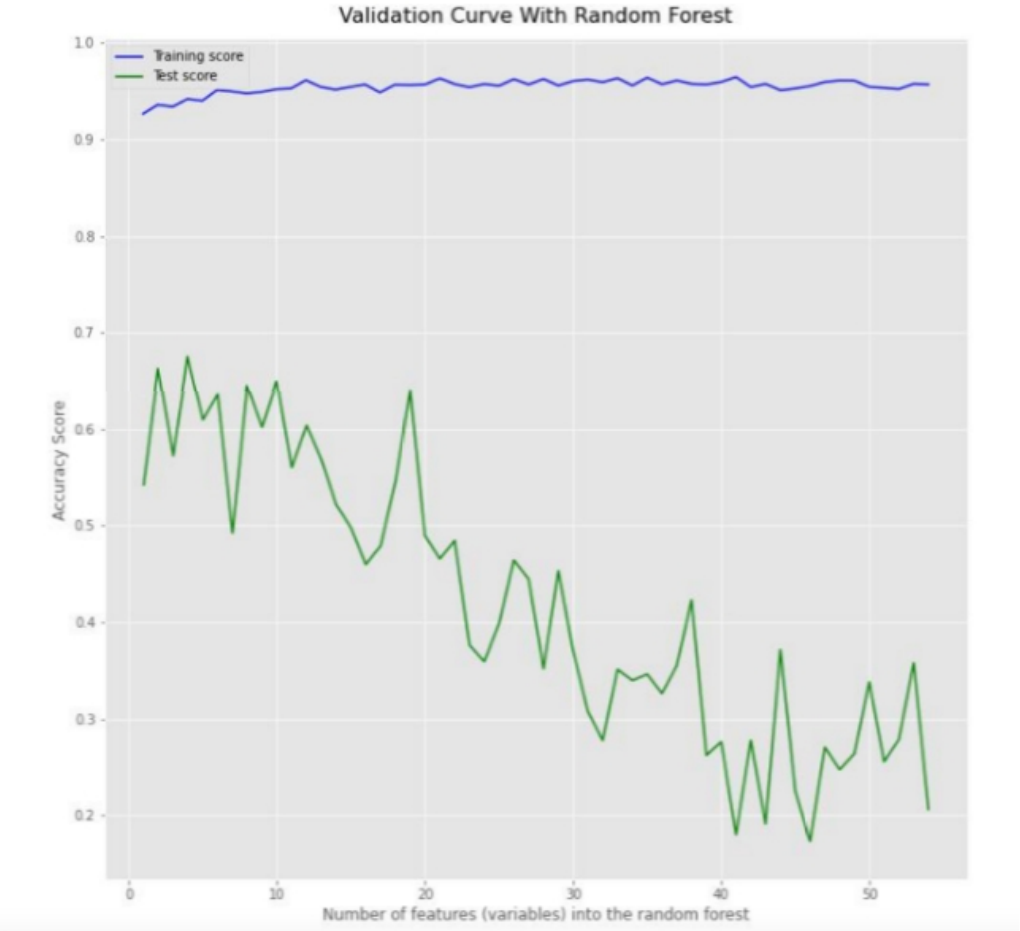
To define our hyperparameters, we use **GridSearchCV().**
To split our dataset, we use **cross_val_score().**

Note about hyperparameters decision

To determine our hyperparameters, we plot a validation curve using an accuracy score from the **scikit-learn library**.
The validation curve is a graph of the decision based on our **GridSearchCV().**

For instance, on this graph, we measure the best *max_features* hyperparameter for our **random forest** model.

We notice the best score returns **max_features = 5** which means that our random forest will have maximum 5 variables into each individual tree.

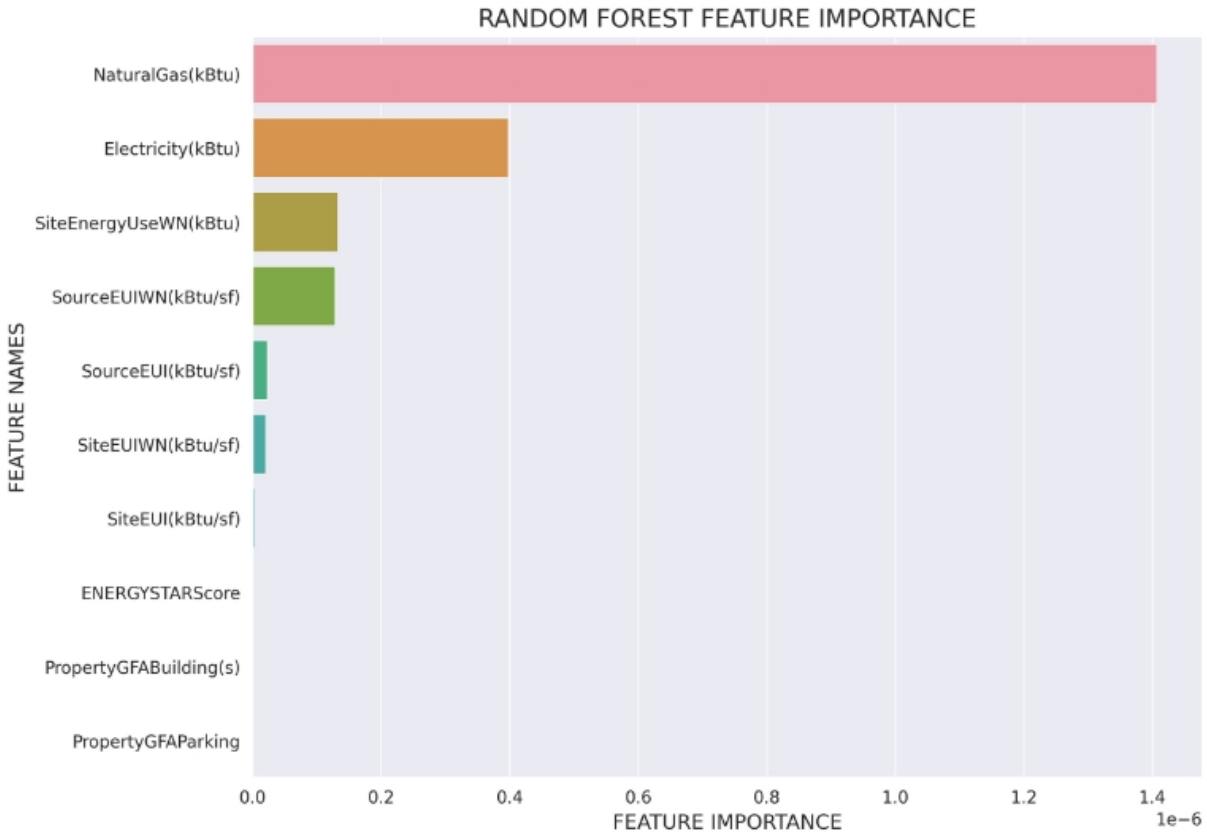


Note about features importance

For our ML models based on decision trees which means:

- **Bagging**
- **Random forest**
- **Decision trees**

We create a graph that returns the features by order of importance.
For instance, here, we notice *NaturalGas(kBtu)* our most important feature for the **random forest** model.



Statistical results #1

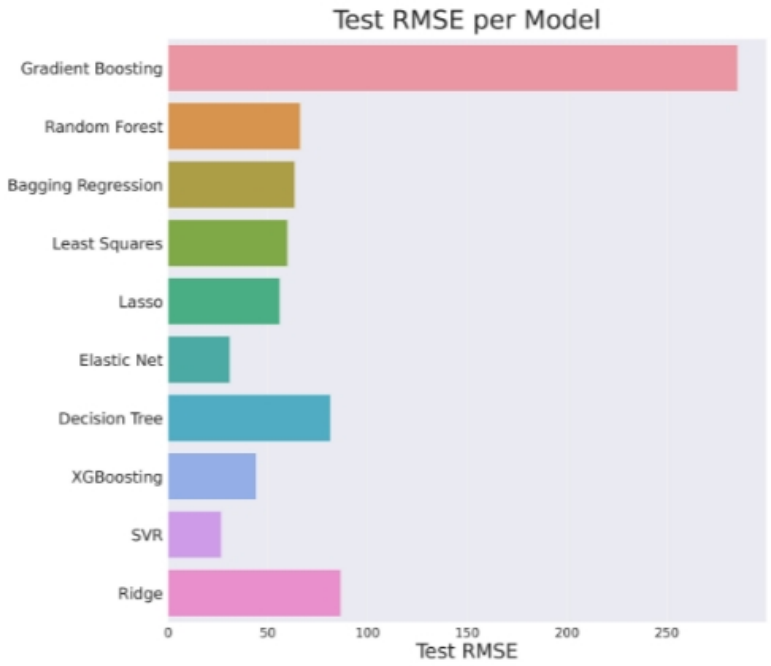
	Method	Elapsed Time	Best Hyperparameter	Training MAE	Test MAE	Training MSE	Test MSE	Training RMSE	Test RMSE	Training R2	Test R2
0	Gradient Boosting	0.366916	{'alpha': 0.85, 'learning_rate': 0.75, 'max_depth': 7, 'max_features': 12, 'max_leaf_nodes': 20, 'min_impurity_decrease': 0.5, 'min_samples_leaf': 4, 'min_weight_fraction_leaf': 0.39, 'n_estimators': 20}	23.859306	22.888637	117554.116134	9882.011514	342.861657	99.408307	0.7765	0.782982

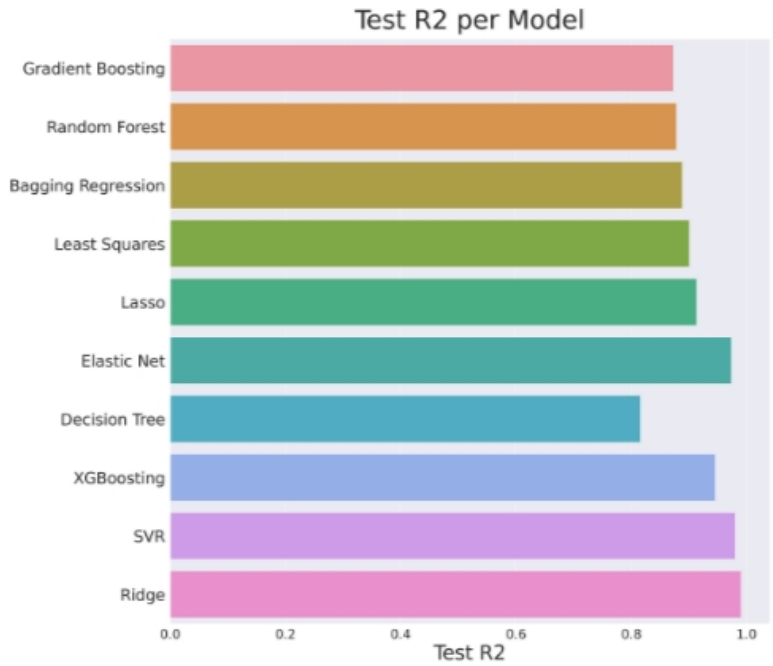
We gather our results for each model in a table where we display:

- **Model**
- **Elapsed time**
- **Hyperparameters**
- **MAE Score for Training and Test sets**
- **MSE Score for Training and Test sets**
- **RMSE Score for Training and Test sets**
- **R2 Score for Training and Test sets**

Statistical results #2

To measure the success of our models, we base our decisions on two main metrics: **RMSE (similar to MSE)** and **R2**.
A **higher R2** gives a better precision of our model. We also want a **lower RMSE** which means a lower standard deviation and variance in our residuals to fit our data.





Final model decision

For both predictive variables, the **Ridge** regression model works best because it returns with the **highest R2** and the **lowest MSE** so we decide to keep this model for our final decision model.

We **normalize** the features of the models using **MinMaxScaler()** and return the weight of each feature per importance when running the Ridge regression model. We can notice the weight of the *ENERGYSTARScore* for each notebook in the next slide.

Features weight for ENERGYSTARScore

TotalGHGEmissions (32 out of 53 features)

16	BuildingType_Multifamily HR (10+)	0.392805
17	Neighborhood_DELRIDGE	0.390397
18	PrimaryPropertType_Low-Rise Multifamily	0.386076
19	Neighborhood_MAGNOLIA / QUEEN ANNE	0.385518
20	Neighborhood_EAST	0.383028
21	Neighborhood_Ballard	0.376595
22	Neighborhood_Central	0.374793
23	Neighborhood_BALLARD	0.367350
24	SiteEUIWN(kBtu/sf)	0.356344
25	SiteEnergyUseWN(kBtu)	0.353176
26	NaturalGas(kBtu)	0.353173
27	Electricity(kBtu)	0.353173
28	PropertyGFABuilding(s)	0.353172
29	PropertyGFAParking	0.353171
30	SiteEUI(kBtu/sf)	0.352914
31	Neighborhood_North	0.350920
32	ENERGYSTARScore	0.348963
33	Neighborhood_SOUTHEAST	0.335754
34	Neighborhood_SOUTHWEST	0.334392

SiteEnergyUseWN (21 out of 53 features)

6	Neighborhood_North	0.677516
7	PrimaryPropertType_Warehouse	0.676956
8	SourceEUIWN(kBtu/sf)	0.672581
9	Neighborhood_Northwest	0.671409
10	PrimaryPropertType_Supermarket / Grocery Store	0.668521
11	PrimaryPropertType_Medical Office	0.634284
12	BuildingType_Campus	0.572910
13	PrimaryPropertType_Mixed Use Property	0.531946
14	PrimaryPropertType_Retail Store	0.525176
15	BuildingType_Nonresidential COS	0.497688
16	BuildingType_Multifamily MR (5-9)	0.488953
17	Neighborhood_NORTHWEST	0.486971
18	Neighborhood_NORTH	0.483929
19	Neighborhood_DELRIDGE	0.441969
20	TotalGHGEmissions	0.438604
21	ENERGYSTARScore	0.419157
22	PropertyGFAParking	0.414402
23	Electricity(kBtu)	0.414401

Questions?