Appliances Energy Prediction

Table of Contents

1. Introduction:

In the pursuit of a more energy-efficient future, the initial step towards it involves the understanding of energy consumption patterns and the ability to predict them accurately. This project is dedicated to the first initial step as its objective. Using the power of data analysis and predictive modeling, we are going to find insights by shedding light on consumption patterns, identifying and following the trends, and ultimately anticipating and predicting energy consumption based on these patterns and trends.

2. Exploratory Data Analysis (EDA):

2.1 About the Dataset:

Title - Appliances Energy Prediction

Source – UCI Machine Learning Repository

Link - https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction

Summary –

The data was taken every 10 minutes for 4.5 months from a house in Belgium. The house consists of 7 rooms and 1 bathroom. The dataset has temperature and humidity collected from inside and outside of the building via sensors and weather stations respectively.

2.2 Data Cleaning:

The data preparation process is essential for ensuring that the dataset is fit for analysis and compatible with regression models. We perform various data cleaning processes such as converting datatype, creating new columns, selecting subsets of columns to form subsets, and transforming data. The following steps are what we followed for effectively cleaning and structuring our data:

1. Converting Date Column:

- The date column is converted into a datetime variable to help us with the next step.

2. Splitting Date Column:

- The date column is split into separate columns for day and month, omitting the year as it remains constant throughout (2016).
- We perform this for 2 major reasons, one is that it helps us with building data visualizations and the other is that regression models don't accept datetime columns.

3. Separating Time:

- The time component is extracted from the date column and placed into a separate column 'time'.

4. Time Column Transformation:

- We now transform the time column into 2 columns – hours and minutes. We have excluded seconds, as the data is collected every 10 minutes.
- This is useful for both building data visualizations and for the regression models.

5. Calculating Total Energy Consumption:

- A new column, 'total,' is calculated by summing the energy consumption of appliances and lights, which helps us understand the total energy consumption.

6. Dropping Redundant Columns:

- Columns such as the date column, time, appliances, and lights columns are now redundant since we have moved the information from those columns into other newer columns.
- Hence, they can be removed to help align the dataset with the requirements of regression models.

7. Dropping Uninformative Columns:

- Columns 'rv1' and 'rv2' are dropped from the dataset, given the absence of any useful information about these columns.

8. Creation of Daily Data Subset:

- A subset of the data is generated by aggregating values for each day by summing the data points every 10 minutes to create the daily data dataset.

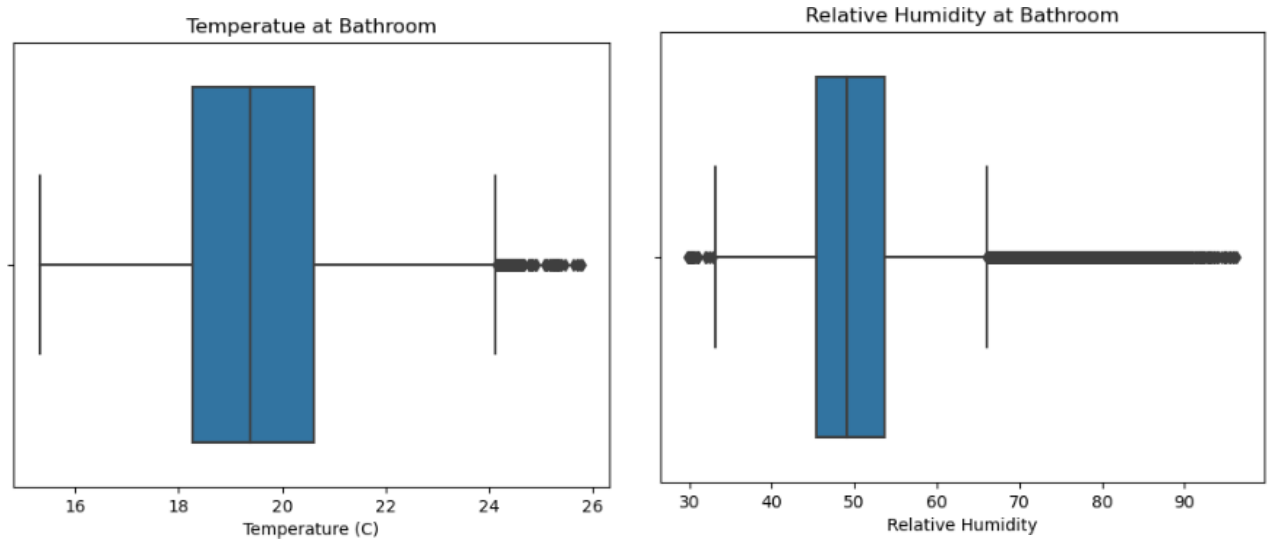
9. Additional Day-of-Week Column:

- The 'day_of_week' column is introduced, which helps us with some data visualizations.

This meticulous data preparation lays the groundwork for meaningful analysis while also improving the dataset's compatibility with regression models.

2.2.1 Outlier Analysis:

Outliers can be defined as the unusual values in our dataset. Outliers have the potential to distort the statistical analysis and assumptions but can also contribute to providing valuable insights. The decision of how to address outliers depends on the nature of the data, and if chosen to be removed then their removal needs to be justified. With the help of box plots, we can identify these outliers and try to make a decision on how to handle them.



The visualizations highlight numerous outliers, but on closer examination, such as comparing attributes like T5 and RH_5, provides meaningful context. Taking the example of temperature (T5) and relative humidity (RH_5) recorded from the bathroom, outliers with a temperature above 24°C and relative humidity above 60% are deemed reasonable. Recognizing that these instances fall within a specific category, it was decided not to remove these outliers, as they contribute valuable insights to the dataset.

Additionally, thorough checks for outliers on various other variables also resulted in the same decision that it was reasonable and didn't need to be removed.

2.2.2 Missing Value Analysis:

We check for missing values in our dataset and identify that our dataset does not contain any missing values.

```
df.isnull().any()
```

```
T1            False
RH_1          False
T2            False
RH_2          False
T3            False
RH_3          False
T4            False
RH_4          False
T5            False
RH_5          False
T6            False
RH_6          False
T7            False
RH_7          False
T8            False
RH_8          False
T9            False
RH_9          False
T_out         False
Press_mm_hg   False
RH_out        False
Windspeed     False
Visibility    False
Tdewpoint     False
day           False
month         False
total         False
hour          False
minute        False
dtype: bool
```

2.2.3 Check for Duplicates:

Now, we check the dataset for duplicates and find out that our dataset does not contain any of them.

```
[ ]: #to check if duplicates
df_duplicates = df[df.duplicated()]
df_duplicates
#no duplicates found ------- to con
```

	date	Appliances	lights	T1	RH_1	T2

0 rows × 29 columns

Now data cleaning is complete, and we can start performing statistical descriptive analytics, data visualizations, and building predictive models on this data.

2.3 Statistical Descriptive Analysis:

As part of Statistical Descriptive Analysis, we find out the following:

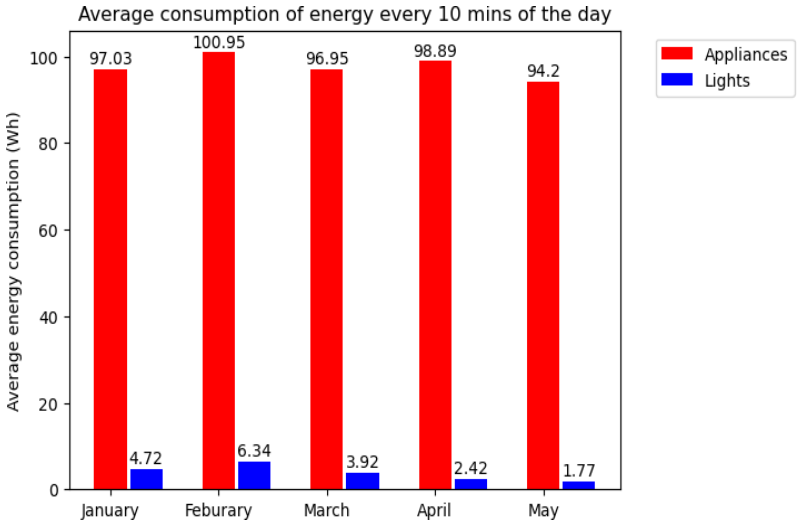
- Measure of central tendency (mean, median, mode)
- Measure of dispersion (standard deviation, minimum, maximum)
- Quartile ranges (25%, 50%, 75%)

	Appliances	lights	Total
count	138.000000	138.000000	138.000000
mean	13971.086957	543.695652	14514.782609
std	4393.151220	457.535569	4531.328613
min	5400.000000	0.000000	5400.000000
25%	10812.500000	172.500000	11537.500000
50%	13255.000000	450.000000	13920.000000
75%	16020.000000	777.500000	16272.500000
max	27150.000000	2180.000000	27690.000000

We have used this data as part of our analysis, in the upcoming data visualizations.

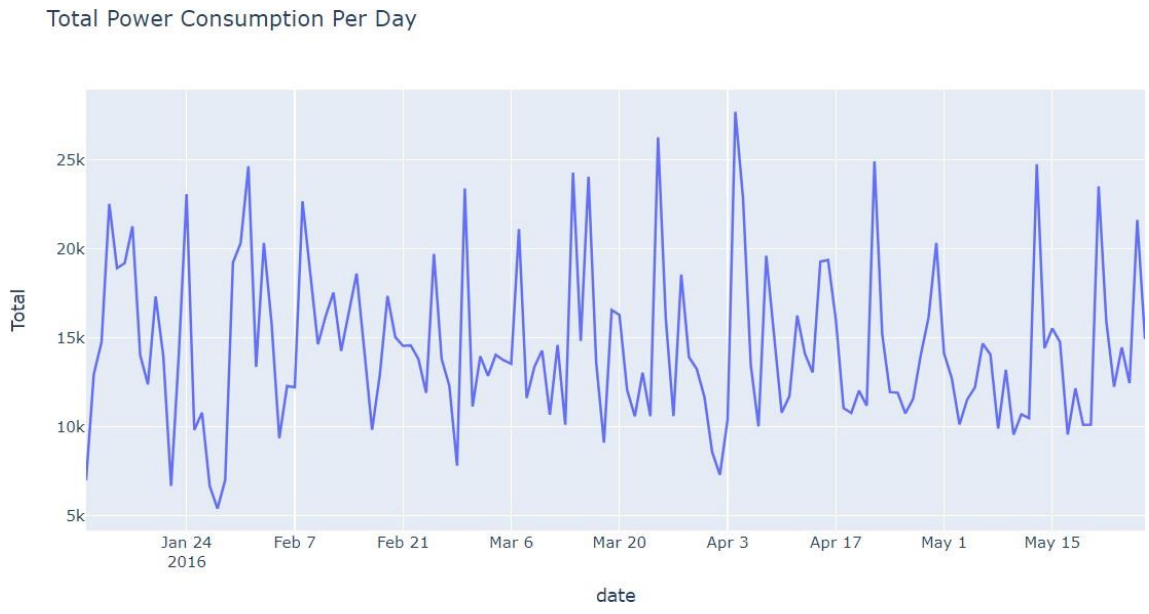
2.4 Data Visualizations:

The dataset consists of data collected at 10-minute intervals, so let's analyze it for average energy consumption across different months and offer insights into temporal patterns and trends.



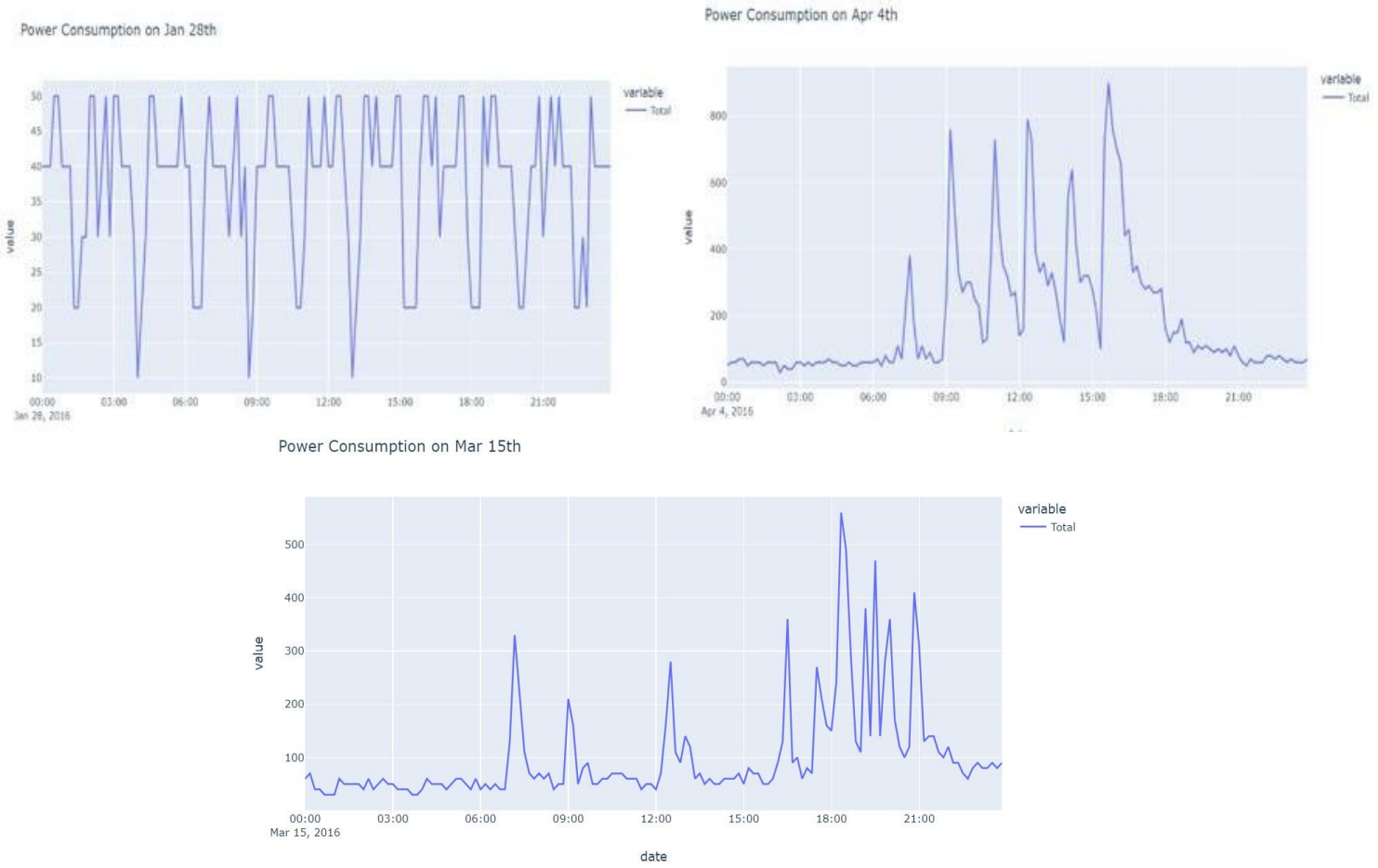
To gain even more valuable insights, we plan to use the aggregated data of per-day energy consumption. We have already created the required aggregated data subset as part of our data-cleaning process. This allows us to gain a better understanding of the daily energy consumption in the given house.

2.4.1 Power Consumption per day and comparative analysis:



Analyzing the daily total energy consumption helped us reveal interesting insights such as January 28th has the minimum energy usage and April 4th recorded the maximum consumption. This insight sheds light on significant variations in energy demands on specific days.

Comparative analysis of average day energy consumption data, using March 15 as a reference as the electricity consumption is almost equal to the mean electricity consumption value, helps understand factors driving extreme energy consumption on specific dates, providing insights into distinct characteristics.



Let's first try to understand the patterns on a normal energy-consuming day. Analyzing the data of March 15th reveals interesting daily trends. As it falls on a Tuesday, a spike in energy usage is observed around 7-8 a.m., likely due to morning routines and preparing to leave for work or school. Another surge around 1 p.m. is noticed and probably during lunch preparation, following this is a steady rise around 6 p.m. as the family members return home from work or school. Finally, the energy consumption gradually declines beyond 9 p.m., indicating that might be around bedtime.

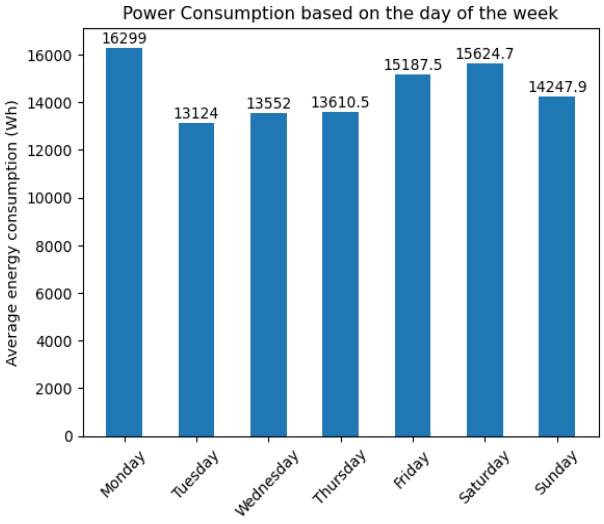
When we try to analyze the trends behind the minimum consumption day on January 28th, a pattern with clues is identified. Throughout the day, energy usage fluctuates between 10-50 Wh. The same pattern could be observed on the March 15th graph, where during sleeping hours when no one is active, a consistent baseline consumption around the same value is observed. The only possible explanation for this would be that electrical appliances refrigerators, cooling/heating units, TVs, and

any other appliances connected to the electricity will draw some energy constantly. With this as evidence, we can suggest that the house was empty on January 28th.

Unfortunately, the graph of April 4th presents us with a puzzle. The high consumption makes us believe that all the members of the family are at home and involved in some heavy energy-consuming tasks for the whole day. On the contradiction, it was a Monday and not a holiday which suggests that it was highly unlikely to be the reason. Energy consumption between 9 a.m. and 6 p.m. exhibits periodic spikes every 2 hours. The lack of a clear explanation raises questions regarding the factors that contributed to this peak consumption.

2.4.2 Weekly Power Consumption Analysis:

Next, we try to analyze the power consumption based on the day of the week. This might help identify trends and fluctuations within a week, additionally giving us valuable insights into behavioral patterns followed by the family.



The analysis of weekly power consumption patterns reveals the following insights:

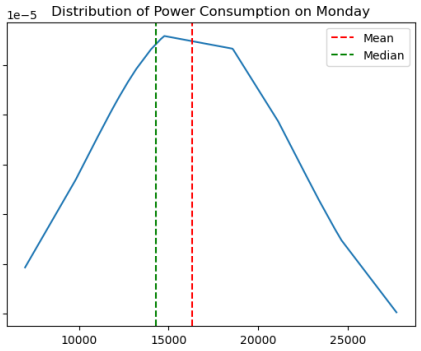
- Mondays exhibit the highest average energy consumption, which might be because of higher activity as all the family members resume their weekday routines.
- Mid-week, from Tuesday to Thursday, a dip in energy usage can be witnessed. This is likely due to fatigue, which results in decreased engagement in entertainment activities or household activities.
- Fridays and Saturdays experience an increase in consumption, reflecting increased entertainment and household activities as the time of the week to relax has begun.
- Sunday sees a decrease in consumption, which aligns with expectations of people tending to head out or just relax at their homes before the next week begins.

These insights help us to try to form a behavioral understanding of the family's weekly pattern and activities.

While all of the analysis insights seemed to fall into place, we had a different feeling about Monday's high mean value. To confirm this, we conducted additional analysis to see if the large difference between Monday and the other days of the week is truly the case.

We use descriptive statistics on the data to look at the median value, because mean values can be easily skewed.

	Total							
	count	mean	std	min	25%	50%	75%	max
Monday	20.0	16299.000000	5850.512258	7000.0	12022.5	14295.0	21490.0	27690.0
Tuesday	20.0	13124.000000	3282.676059	9550.0	10765.0	12665.0	14310.0	22810.0
Wednesday	20.0	13552.000000	3695.347858	6660.0	11815.0	13140.0	14232.5	24050.0
Thursday	20.0	13610.500000	4087.671419	5400.0	11032.5	12975.0	14662.5	22520.0
Friday	20.0	15187.500000	5950.745973	6680.0	9930.0	14800.0	19002.5	26260.0
Saturday	19.0	15624.736842	3572.262539	7300.0	14075.0	15040.0	17870.0	23500.0
Sunday	19.0	14247.894737	4132.690244	7810.0	10695.0	14120.0	16150.0	23070.0

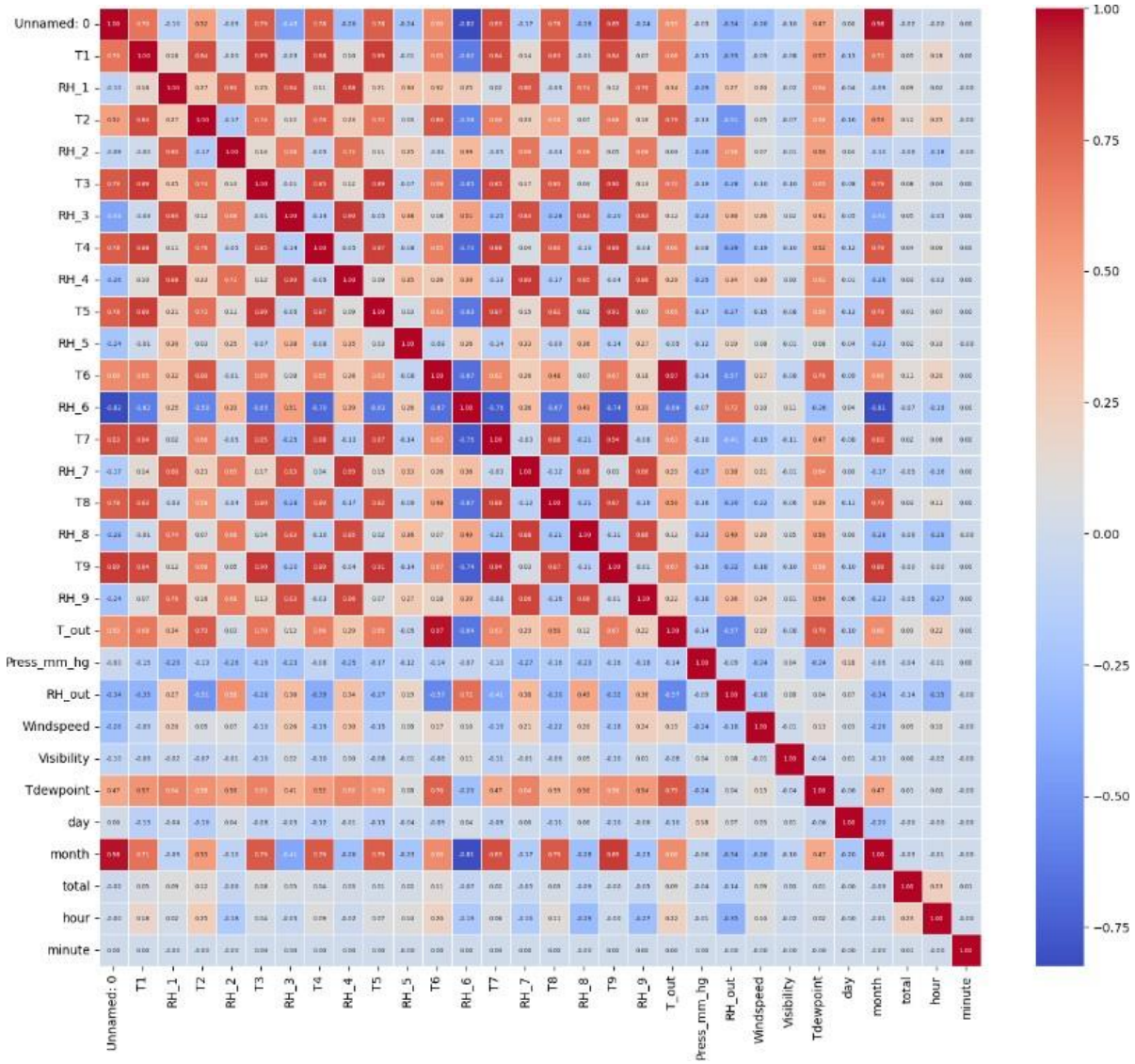


As we can see, Monday's median value is nearly equal to the other days and is not as high as the mean value portraits. Because April 4th (Monday) has the highest recorded energy consumption, this large value causes the mean to skew towards a larger value.

The rest of the inferences made using mean values are matching with the patterns observed using the median values.

2.4.3 Correlation Analysis:

Correlation analysis is used to uncover linearity within the dataset. The observations derived from the correlation heat map indicate the degree of association between variables. These insights help in identifying patterns, dependencies, and potential predictors in data, laying the groundwork for subsequent analyses and model development.



Inferences:

1. The majority of the variables are not correlated to other variables.
2. The temperature outside the house is correlated with the temperature inside the house. Similarly, the temperatures between the different rooms are also correlated.
3. Looking at additional weather data, we can infer that windspeed and visibility are not correlated to the temperature inside the house, but the dew point is correlated.
4. The month variable also shows some correlation with the temperature of different rooms, which is understandable as when the season changes the weather also changes.

We can conclude from the correlation analysis that there is not enough correlation between the variables indicating that most variables are independent of each other. So overall, we can infer that the data has no visible linearity between the variables.

3. Train – Test Dataset Creation:

The dataset is now divided into training and testing sets. This is a fundamental step in predictive modeling. We perform an 80-20 split, such that 80% of the total data is allocated to the training set and the remaining 20% becomes the test set.

The training set serves as the foundation for model training and the test set serves as the metric for evaluating the performance. By evaluating the predictive model on unseen test data, we make sure that the model's performance is reliable.

The number of data points for each of the sets is as follows,

Training Data	Testing Data
15788	3947

4. Predictive Models:

Before we start building predictive models, it is important to find out the following things.

1) What is the prediction label? – **Total (Power Consumption)**
2) What is the prediction task? – **Regression**

Now, let's try to fit and compare various regression models to see how they perform on this dataset.

4.1 Linear Regression

Linear regression is a statistical method that fits a linear equation to observed data to model the relationship between a dependent variable and one or more independent variables. We experimented with three different linear regression models - linear regression, lasso linear regression, and ridge linear regression. Let's compare the regression model's performance.

Models	R-Squared	Mean Absolute Error
Linear Regression	0.158	54.68
Lasso Regression	0.155	54.34
Ridge Regression	0.158	54.68

Linear regression models certainly failed for our data since none of the variables are correlated to each other. As identified from our correlation analysis the dataset has no linearity, hence a linear regression model will not be a good fit for this type of data.

4.2 SVM (Support Vector Machine)

Next, we tried SVM as it has an advantage over linear regression as kernels help introduce flexibility to model non-linear decision boundaries. We used different types of kernels, to better understand the performance between them. Let's compare the performance of various kernel SVM's.

SVM		
Kernel	**R-Squared**	**Mean Absolute Error**
Linear	0.049	45.058
Polynomial	-0.093	49.965
RBF	-0.105	50.588
Sigmoid	-0.133	51.609

SVM with a linear kernel, like linear regression, is ineffective due to the non-linear nature of the data. Attempts to introduce non-linearity using different kernels produce unsatisfactory results, leading to the conclusion that SVM is not appropriate for the dataset in question.

4.3 Random Forest

Random forest is a bagging algorithm, that trains on multiple models on different subsets of the dataset, and prediction is based on the average of predictions (since regression task). So, we try to change the parameter n_estimator to train it on fewer or more models to see how the performance varies. Let's compare the performance of the random forest model.

Random Forest		
n_estimators	**R-Squared**	**Mean Absolute Error**
10	0.556	32.34
50	0.609	30.332
100	0.614	30.115
200	0.619	29.91

We can easily see that Random Forest has a high R-squared value, indicating that it performs quite well. We can see that increasing the n_estimators from 10 to 200 improves the performance of the regression model. This is the first model that works well on our dataset and beats the before models by far.

4.4 XGBoost

XGBoost is a famous machine-learning model that works by creating ensembles of decision trees using a gradient-boosting framework. XGBoost is well known for its regularization, and feature importance techniques.

We try changing the following parameters to see a difference:

- Booster: Booster parameter sets the type of learner - linear or tree-based function.
- N_estimators: Number of models to be built on

Let's compare the various XGBoost model's performance

XGBoost (booster='dart')		
n_estimators	**R-Squared**	**Mean Absolute Error**
10	0.127	40.203
100	0.359	34.202
200	0.403	33.225
500	0.448	32.432
XGBoost (booster='gblinear')		
100	-0.07	49.44

Since Random Forest performed well, we had hopes that XGBoost would perform the same but it fell short in performance. XGBoost with booster 'gblinear' performs poorly, as seen previously with linear models, but 'dart' performs much better because it follows a tree structure similar to a random forest. As can be seen, increasing the number of estimators improves performance.

4.5 KNN

As seen before, the points are clustered together in tight groups, K-NN regression might be a good choice as it adapts well to non-linear relationships and requires minimal assumptions about data distribution. Additionally, it predicts the outcome by averaging the values of its k-nearest neighbors (since it's a regression task).

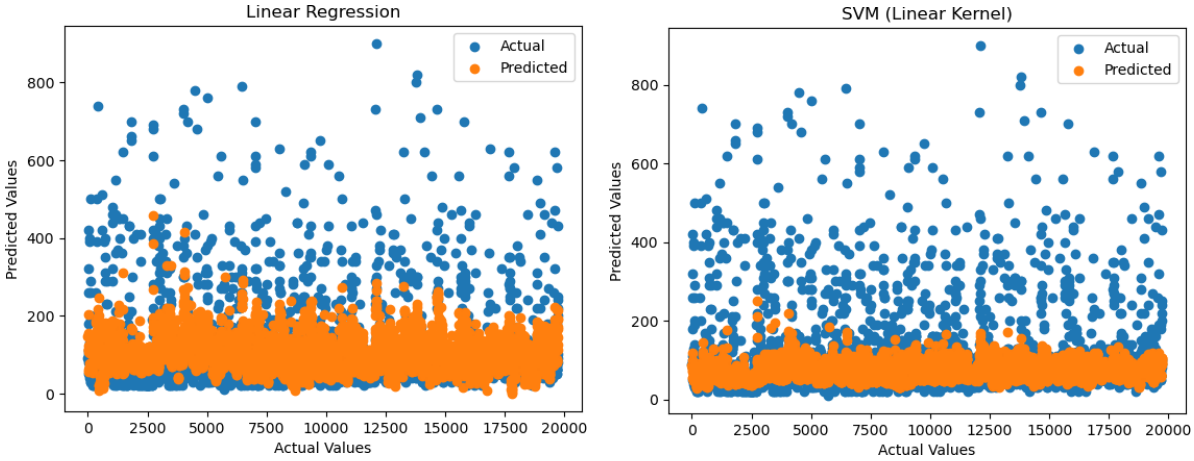
Let's set different k value and compare the model,

KNN		
k	**R-Squared**	**Mean Absolute Error**
5	0.287	45.758
10	0.257	47.974
25	0.198	51.448
50	0.159	53.685

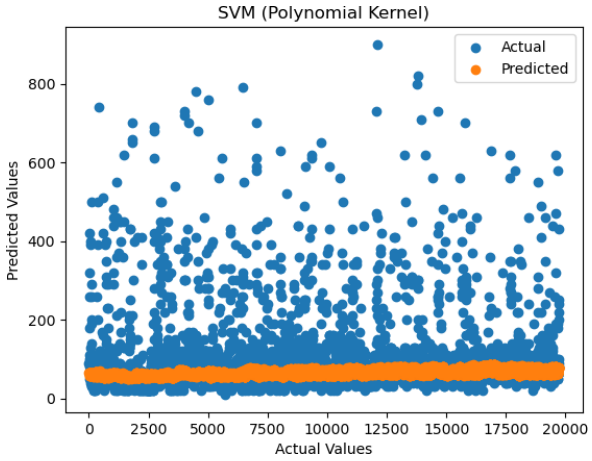
The KNN model outperforms the linear and SVM models, but not the Random Forest or XGBoost models. We can see that reducing the number of neighbors helps raise the R-squared score, but not sufficiently.

5. Conclusion

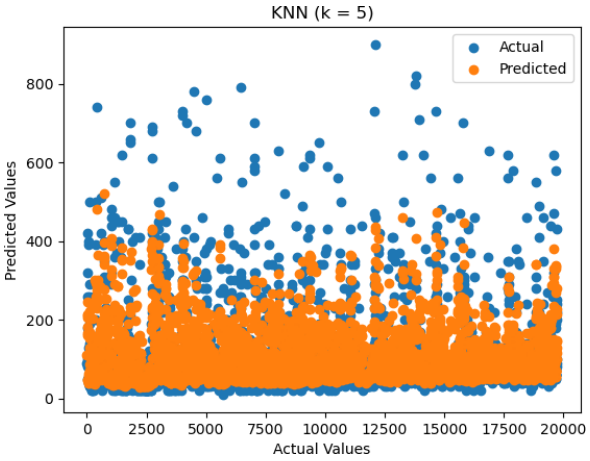
In summary, when applied to a dataset with non-linear relationships, linear regression and SVM models with linear kernels are ineffective. As seen below, it tries to fit a linear line which is not sufficient to capture the spread of the data.



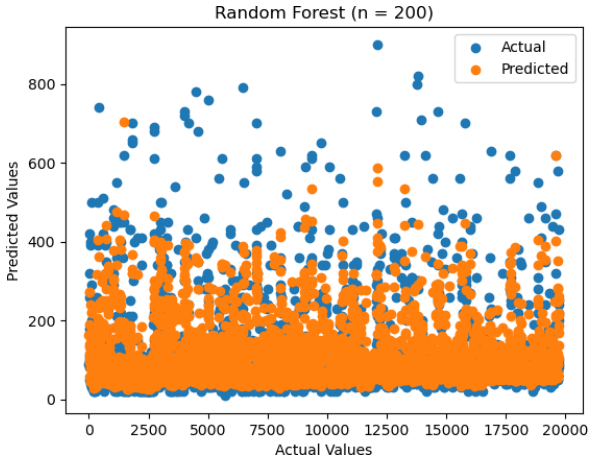
SVM kernels which we expected to perform well for non-linear relationships didn't even perform as well as the linear regression models. The reason for this is that the kernels were trying to fit a linear line for some reason and is tighter than the linear kernel of SVM.

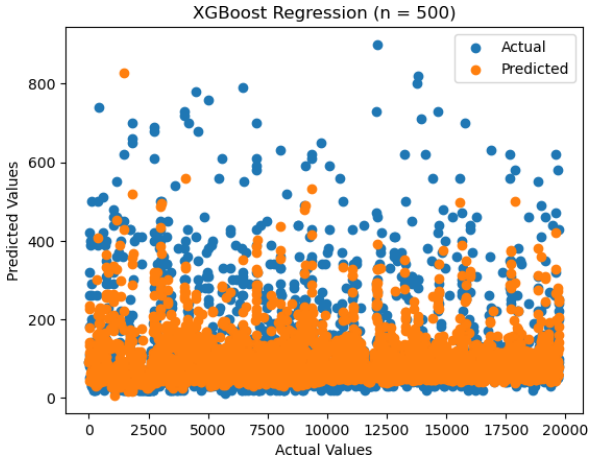


The performance of KNN was better than linear models but they were not satisfactory. From the graph below it seems to try capture the data but somewhere it is failing in doing it correctly which is the reason for the low performance.



Random Forest consistently outperforms other models, with its predictive power improving as the number of estimators increases. XGBoost with 'dart' also provides competitive results.





We can see that Random Forest and XGBoost try to capture the data points and trends and have more flexibility.

In conclusion, for this dataset, Random Forest and XGBoost with 'dart' are the top-performing models, emphasizing the importance of selecting models tailored to the data's characteristics. Linear models and SVM with linear kernels are not suitable for such non-linear data.

References

1. Candanedo,Luis. (2017). Appliances energy prediction. UCI Machine Learning Repository. https://doi.org/10.24432/C5VC8G.
2. Plotly Express. (n.d.). Plotly.com. https://plotly.com/python/plotly-express/
3. scikit-learn. (2019). scikit-learn: machine learning in Python. Scikit-Learn.org. https://scikit-learn.org/stable/
4. Introduction to Seaborn - Python. (2020, May 31). GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-seaborn-python/
5. XGBoost Parameters — xgboost 1.5.2 documentation. (n.d.). Xgboost.readthedocs.io. https://xgboost.readthedocs.io/en/stable/parameter.html