Capstone Project - 3

Appliances Energy Prediction

Team Members

Amrutha B S
Mahima Shree
Purnima Rai
Pooja Rana
Tanmay Bohra
Vikram Pratap

Index

- Problem Statement
- Data Description
- Data Wrangling
- Exploratory View – All Features, Target Variable
- EDA
- Correlation Matrix
- Feature Engineering
- Data Splitting & Standardizing
- Model Training & Hyper-Parameter Tuning
- Model Performance Comparison
- Conclusion
- Scope of Improvement

Problem Statement

- In this time of global variability world needs energy to support economic and social progress and build a better quality of life, in particular in developing countries. But even in today's time there are many places especially in developing world where there are outages. In this project we will be analyzing the appliance usage in the house gathered via home sensors. All readings are taken at 10 minutes intervals for 4.5 months . The goal is to predict energy consumption by appliances .

Goal

- The goal of our project is to predict the energy consumption of appliances in households based on the sensor data we have and corresponding weather reports.

Data Description

```
temp = {
    'T1' : 'kitchen_temp', 'T2' : 'living_temp', 'T3' : 'laundry_temp',
    'T4' : 'office_temp', 'T5' : 'bath_temp', 'T6' : 'outside_temp',
    'T7' : 'ironing_temp', 'T8' : 'teen_temp', 'T9' : 'parents_temp', 'T_out' : 'station_temp'
}
```

```
humid = {
    'RH_1' : 'kitchen_humid', 'RH_2' : 'living_humid', 'RH_3' : 'laundry_humid',
    'RH_4' : 'office_humid', 'RH_5' : 'bath_humid', 'RH_6' : 'outside_humid',
    'RH_7' : 'ironing_humid', 'RH_8' : 'teen_humid', 'RH_9' : 'parents_humid', 'RH_out' : 'station_humid'
}
```

```
Index(['date', 'Appliances', 'kitchen_temp', 'kitchen_humid', 'living_temp',
       'living_humid', 'laundry_temp', 'laundry_humid', 'office_temp',
       'office_humid', 'bath_temp', 'bath_humid', 'outside_temp',
       'outside_humid', 'ironing_temp', 'ironing_humid', 'teen_temp',
       'teen_humid', 'parents_temp', 'parents_humid', 'station_temp',
       'Press_mm_hg', 'station_humid', 'Windspeed', 'Visibility', 'Tdewpoint',
       'rv1', 'rv2'],
      dtype='object')
```

- The dataset is a series of observations collected from a wireless sensor network from a building in Belgium at an interval of 10 minutes for a period of about 4.5 months.
- The dataset is having 19735 rows and 29 columns.
- The sensor data comprises of temperature and humidity levels in different rooms in the building and outside areas.
- Among other features are weather reports on Pressure, Wind speed, Visibility and T-dew point, which are recorded at weather station Chievres Airport, Belgium.

Data Wrangling

```
# Let us check for duplicates
duplicate = df[df.duplicated()]
duplicate
```

date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	T5	RH_5	T6	RH_6	T7	RH_7	T8	RH_8	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpo

- Checking for duplicates : **None Found**
- Checking for NaN Values : **None Found**
- Dropped Light column : **77% data has zero count**

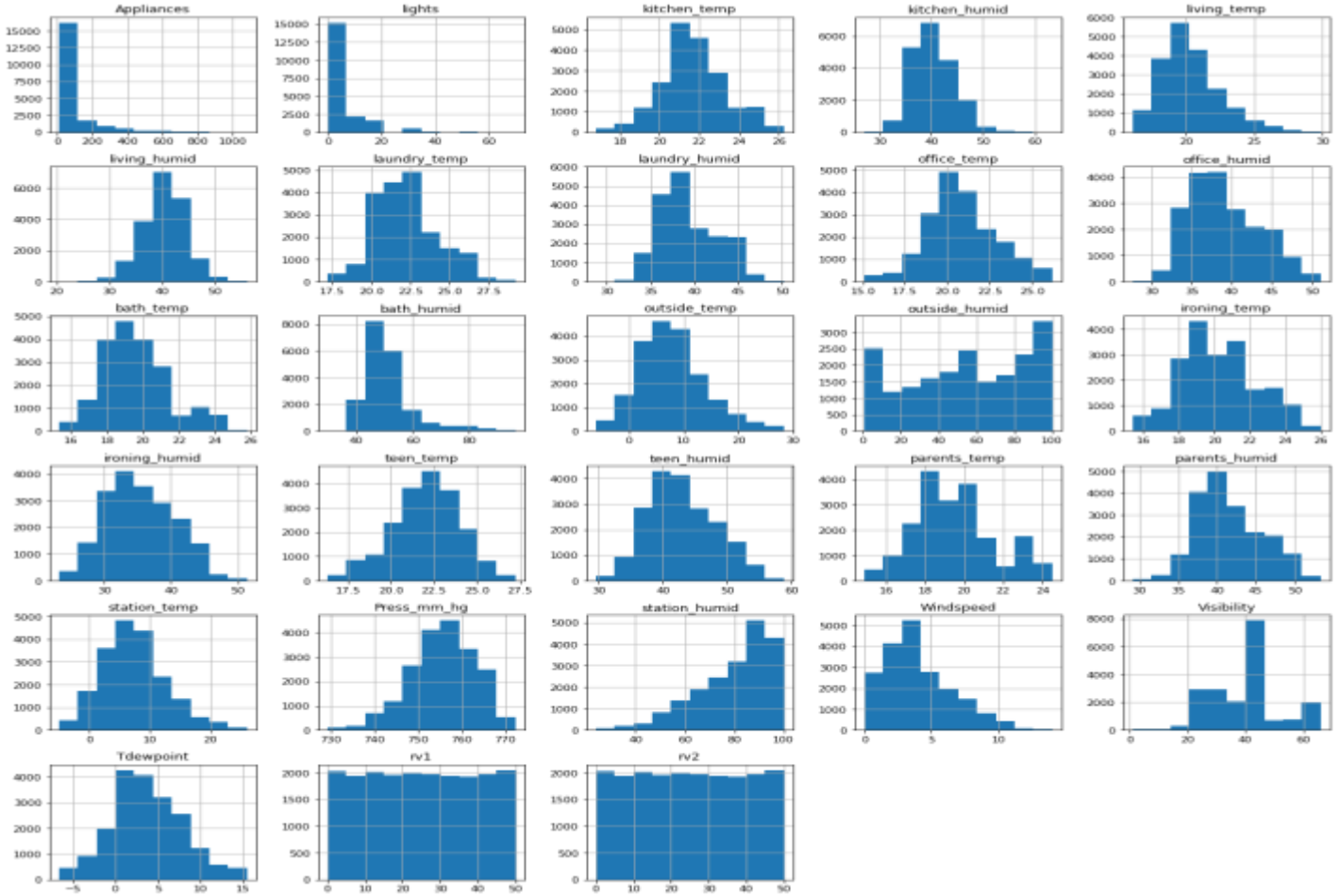
```
df.isnull().sum()
```

```
date           0
Appliances     0
lights         0
T1             0
RH_1           0
T2             0
RH_2           0
T3             0
RH_3           0
T4             0
RH_4           0
T5             0
RH_5           0
T6             0
RH_6           0
T7             0
RH_7           0
T8             0
RH_8           0
T9             0
RH_9           0
T_out          0
Press_mm_hg    0
RH_out         0
Windspeed      0
Visibility     0
Tdewpoint      0
rv1            0
rv2            0
dtype: int64
```

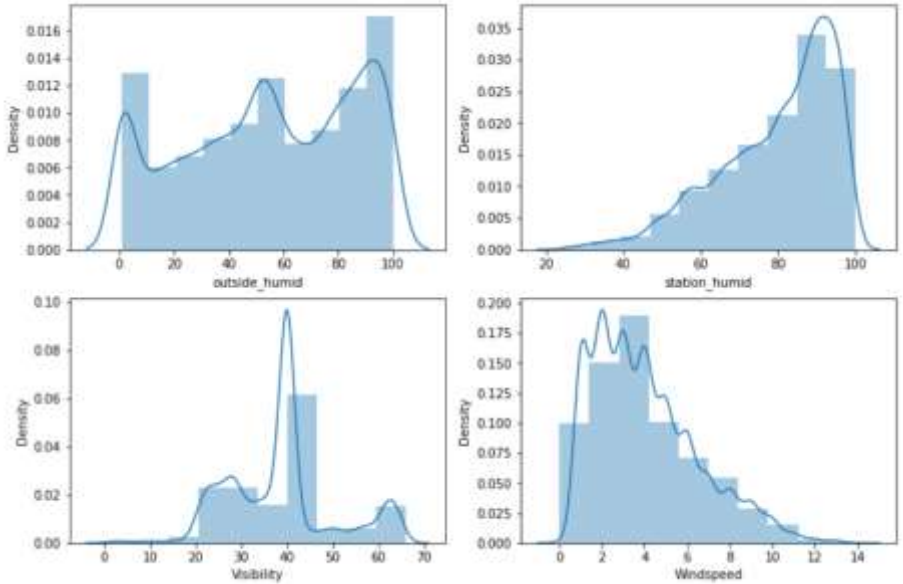
```
df.lights.value_counts()
```

```
0     15252
10     2212
20     1624
30      559
40       77
50        9
60        1
70        1
Name: lights, dtype: int64
```

Exploratory View – All Features



Exploratory View – Irregular Patterns



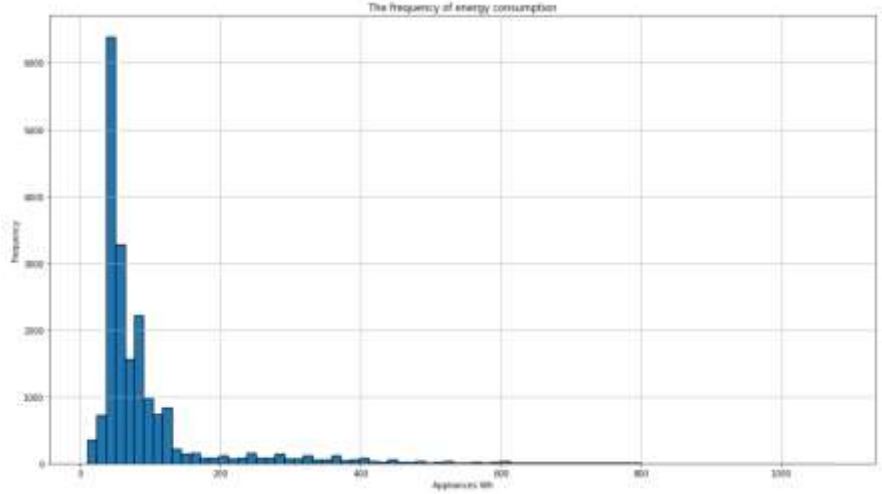
Observations ⚡

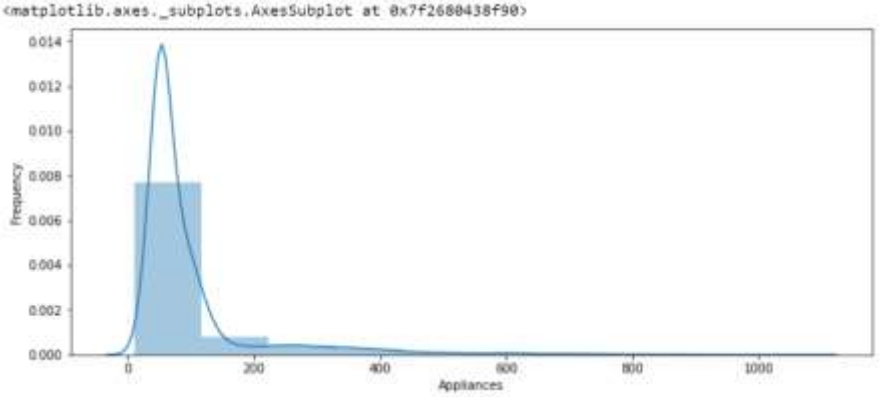
Humidity - All columns follow normal distribution except outside_humid and station_humid , primarly because these sensors are outside the house

Visibilty - This column is negatively skewed

Windspeed - This column is postively skewed

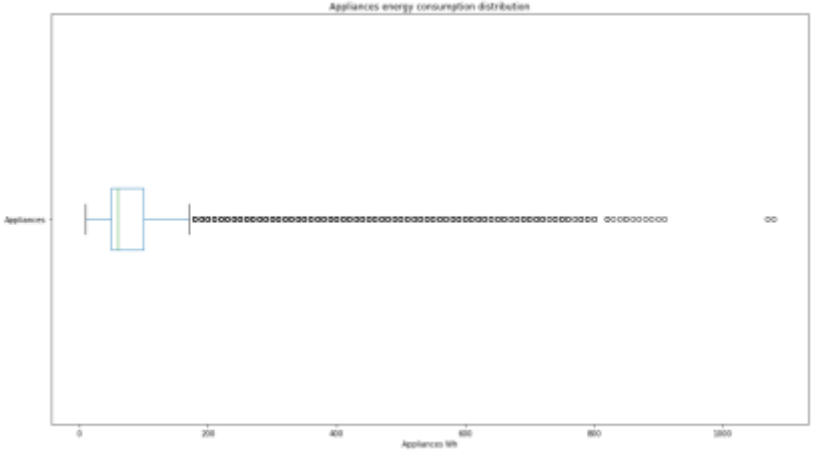
Target Variable - Appliances



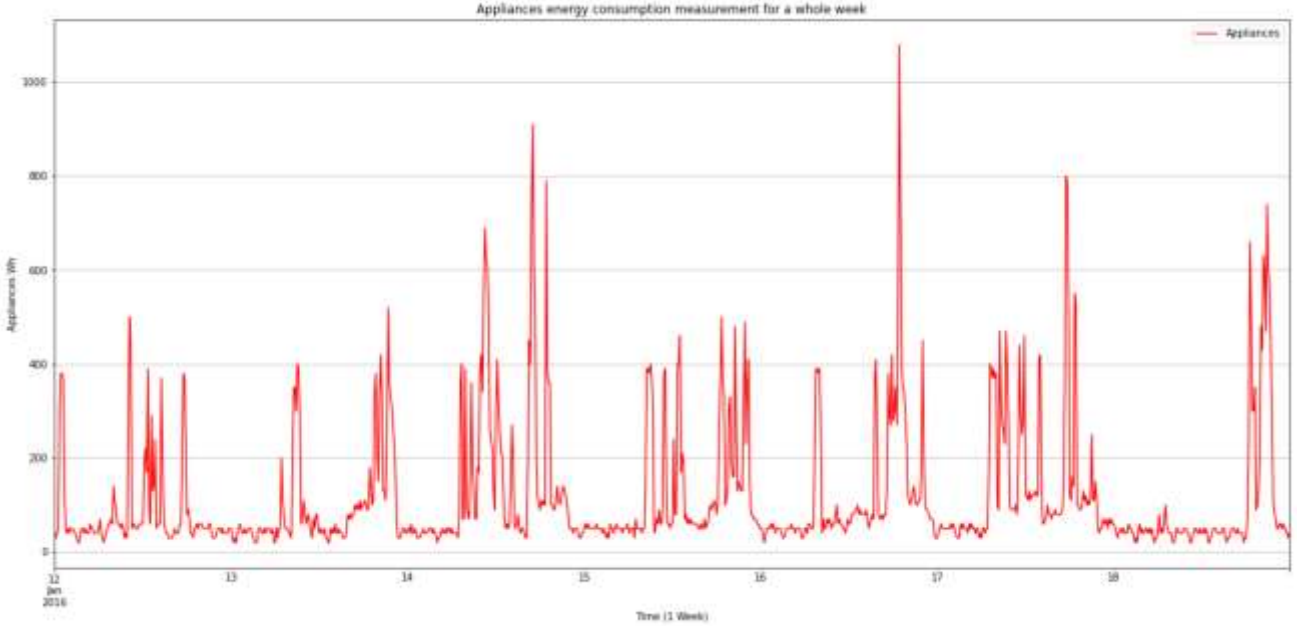


Insights⚡:

1. Energy consumption of appliances ranges from 10 Wh to 1080 Wh.
2. 90% of Appliance consumption is less than 200 Wh .
3. This column is postively skewed , most the values are around mean 100 Wh .
4. There will be outliers in this column.
5. There are small number of cases where consumption is very high.



Target Variable - Appliances



Insights⚡:

1. 16th January 2016, had recorded the highest energy consumption i.e., 1080 Wh.
2. Upon checking the calendar, it was long weekend i.e., Monday 18th January 2016 being a public holiday on account of Martin Luther King Jr. Day.
3. We can see that the consumption was high in the evening time, so there could have been some event and thus it cannot be considered as an outlier.

Feature Variable – Temperature Analysis

```
energy[temp.values()].describe()
```

	kitchen_temp	living_temp	laundry_temp	office_temp	bath_temp	outside_temp	ironing_temp	teen_temp	parents_temp	station_temp
count	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
mean	21.686571	20.341219	22.267611	20.855335	19.592106	7.910939	20.267106	22.029107	19.485828	7.411665
std	1.606066	2.192974	2.006111	2.042884	1.844623	6.090347	2.109993	1.956162	2.014712	5.317409
min	16.790000	16.100000	17.200000	15.100000	15.330000	-6.065000	15.390000	16.306667	14.890000	-5.000000
25%	20.760000	18.790000	20.790000	19.530000	18.277500	3.626667	18.700000	20.790000	18.000000	3.666667
50%	21.600000	20.000000	22.100000	20.666667	19.390000	7.300000	20.033333	22.100000	19.390000	6.916667
75%	22.600000	21.500000	23.290000	22.100000	20.619643	11.256000	21.600000	23.390000	20.600000	10.408333
max	26.260000	29.856667	29.236000	26.200000	25.795000	28.290000	26.000000	27.230000	24.500000	26.100000

Observations⚡:

1. Outside Average temperature over a period of 4.5 months is around 7.5 degrees and ranges from -6(min) to 28(max) degrees.
2. Inside the building avarage temperature has been around 20 degrees for all the rooms and ranges from 14(min) to 30(max) degrees.

Note: These points implies that warming appliances have been used to keep the insides of the building warm. There must be some sort of direct correlation b/w temperature and consumption of energy inside the house.

Feature Variable – Humidity Analysis

```
energy[humid.values()].describe()
```

	kitchen_humid	living_humid	laundry_humid	office_humid	bath_humid	outside_humid	ironing_humid	teen_humid	parents_humid	station_humid
count	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
mean	40.259739	40.420420	39.242500	39.026904	50.949283	54.609083	35.388200	42.936165	41.552401	79.750418
std	3.979299	4.069813	3.254576	4.341321	9.022034	31.149806	5.114208	5.224361	4.151497	14.901088
min	27.023333	20.463333	28.766667	27.660000	29.815000	1.000000	23.200000	29.600000	29.166667	24.000000
25%	37.333333	37.900000	36.900000	35.530000	45.400000	30.025000	31.500000	39.066667	38.500000	70.333333
50%	39.656667	40.500000	38.530000	38.400000	49.090000	55.290000	34.863333	42.375000	40.900000	83.666667
75%	43.066667	43.260000	41.760000	42.156667	53.663333	83.226667	39.000000	46.536000	44.338095	91.666667
max	63.360000	56.026667	50.163333	51.090000	96.321667	99.900000	51.400000	58.780000	53.326667	100.000000

Observations :

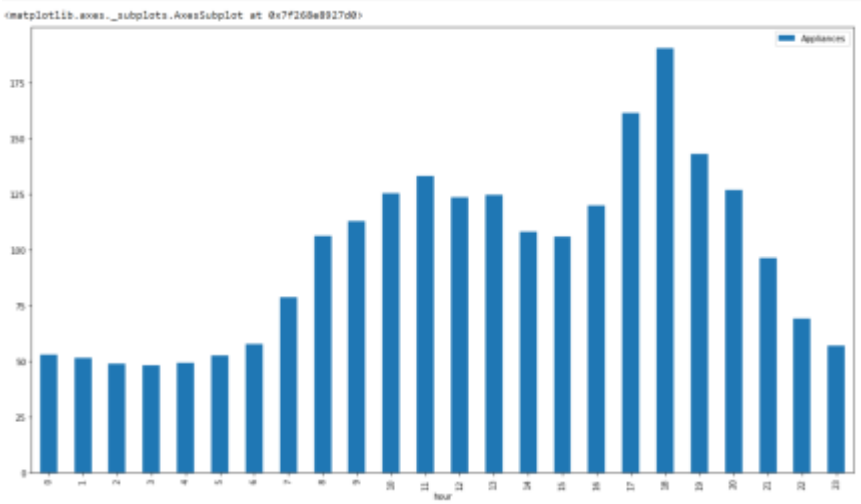
Outside the building average temp **>** average humidity inside the house.

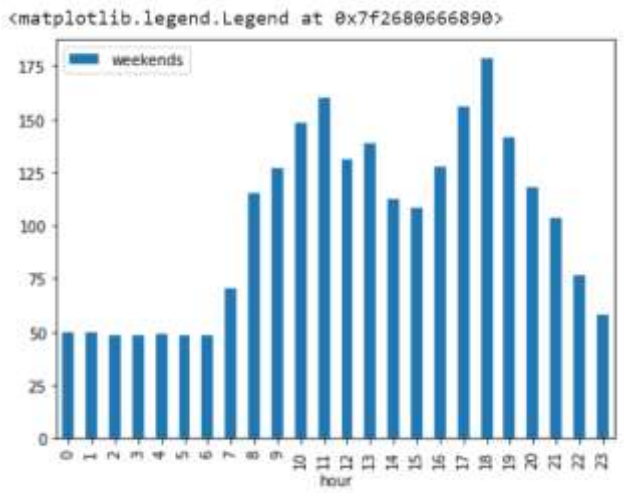
Average humidity at the weather station **>** outside humidity near the building.

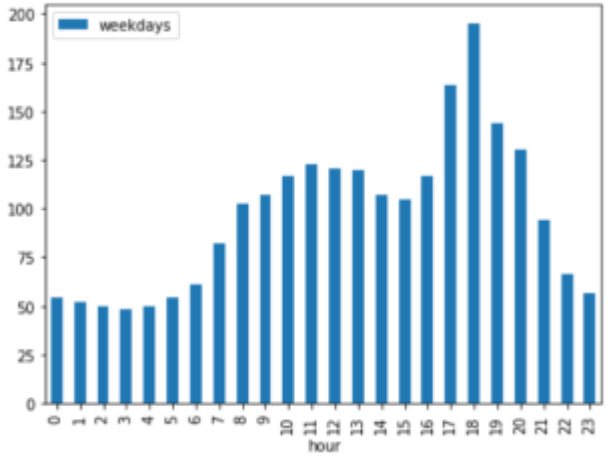
Average humidity in the bathroom **>** other rooms due to obvious reasons.

Kids and parent room show a comparatively higher average humidity.

EDA – Appliances v/s Time



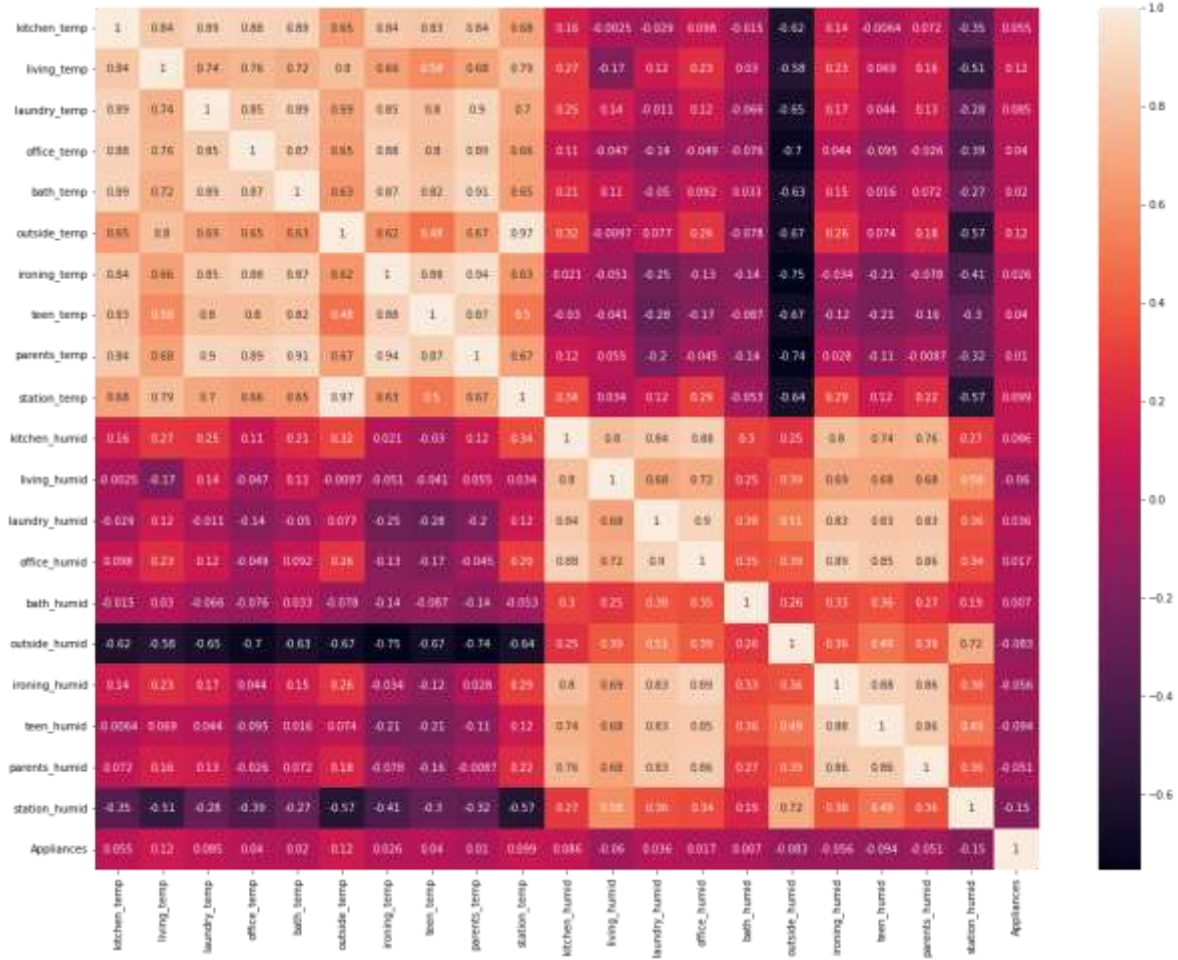




Insights

1. Two peak hours observed - 11 am and 6 pm.
2. Sleeping hours i.e., 10 pm - 6 am, the energy consumption of appliances is around 50 Wh.
3. On comparing the energy consumption on Weekends & Weekdays, we observe higher consumption on weekend throughout the day, especially 8 am to 4 pm.
4. Overall energy consumption in weekends is pretty high.

Co-relation Matrix



- Temperature & Humidity features have strong co-relation among themselves.
- Both these have less co-relation when compared to each other.
- Humidity outside have a strong negative correlation with temperature levels.
- Apart from that we observe that a couple features such as humidity at station, temperature outside the building and temperature in the living room have a comparatively high absolute correlation (above 0.12) with Appliances energy consumption.

Data Splitting, Standardization & Models Used

```
# 75% of the data is usedfor the training of the models and the rest is used for testing
from sklearn.model_selection import train_test_split
train, test = train_test_split(energy,test_size=0.25,random_state=40)
```

```
#Split training dataset into independent and dependent varibales
X_train = train[col_temp + col_hum + col_weather]
y_train = train[col_target]
```

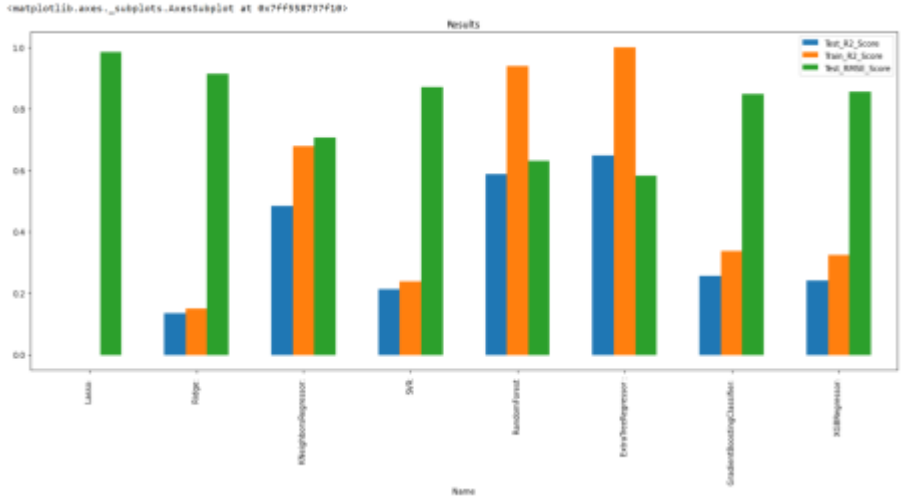
```
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

```
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train.values.reshape([-1,1])).flatten()
y_test = sc_y.transform(y_test.values.reshape([-1,1])).flatten()
```

- Linear Regression
  - Lasso Regression
  - Ridge Regression
- KNN Regression
- Support Vector Machine
- Random Forest Regression
- Extra Tree Regression
- Gradient Boosting Regression

Performance of Models

	Name	Train_Time	Train_R2_Score	Test_R2_Score	Test_RMSE_Score
0	Lasso:	0.009995	0.000000	-0.000031	0.984268
1	Ridge:	0.007406	0.149661	0.134760	0.915535
2	KNeighborsRegressor:	0.002951	0.679698	0.485718	0.705842
3	SVR:	12.601709	0.239142	0.213973	0.872620
4	RandomForest	33.204529	0.939153	0.587949	0.631804
5	ExtraTreeRegressor :	8.236642	1.000000	0.647956	0.583990
6	GradientBoostingClassifier:	7.792264	0.336856	0.257351	0.848200
7	XGBRegressor:	1.701684	0.324972	0.242367	0.856715



Obervations ⚡

1. Best results over test set are given by Extra Tree Regressor with R2 score. of 0.64
2. Least RMSE score is also by Extra Tree Regressor 0.58.
3. Lasso regularization over Linear regression was worst performing model.

Performance of Models – After Hyperparameter Tuning

```
from sklearn.model_selection import GridSearchCV
param_grid = [{
              'max_depth': [80, 150, 200,250],
              'n_estimators' : [100,150,200,250],
              'max_features': ["auto","sqrt", "log2"]
            }]
reg = ExtraTreesRegressor(random_state=40)
grid_search = GridSearchCV(estimator = reg, param_grid = param_grid, cv = 5, n_jobs = -1 , scoring='r2' , verbose=2)
grid_search.fit(X_train, y_train)
```

[ ]
```
# R2 score on training set with tuned parameters

grid_search.best_estimator_.score(X_train,y_train)
```

```
0.9999995875080147
```

[ ]
```
# R2 score on test set with tuned parameters
grid_search.best_estimator_.score(X_test,y_test)
```

```
0.6507922746972608
```

[ ]
```
# RMSE score on test set with tuned parameters

np.sqrt(mean_squared_error(y_test, grid_search.best_estimator_.predict(X_test)))
```

```
59.86441818731427
```

Observations :

Based on parameter tunning step we can see that

Best possible parameter combination are - 'max_depth': 80, 'max_features': 'sqrt', 'n_estimators': 250

Training set R2 score of 1.0 may be signal of overfitting on training set

Test set R2 score is 0.65 improvement over 0.64 achieved using untuned model

Working on Extra-Tree Regressor

```
# Get sorted list of features in order of importance
feature_indices = np.argsort(grid_search.best_estimator_.feature_importances_)
```

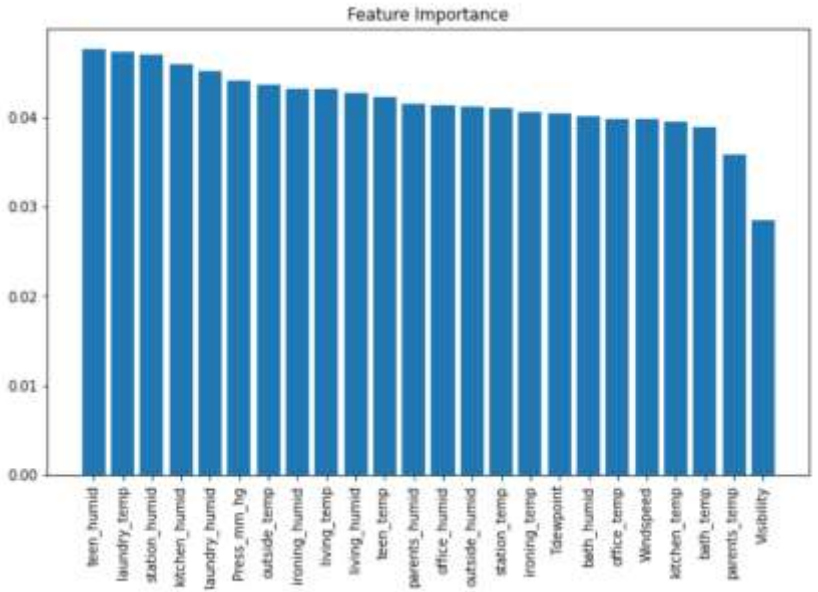
```
importances = grid_search.best_estimator_.feature_importances_
indices = np.argsort(importances)[::-1]
names = [X_train.columns[i] for i in indices]
# Create plot
plt.figure(figsize=(10,6))

# Create plot title
plt.title("Feature Importance")

# Add bars
plt.bar(range(X_train.shape[1]), importances[indices])

# Add feature names as x-axis labels
plt.xticks(range(X_train.shape[1]), names, rotation=90)

# Show plot
plt.show()
```



Observations ⚡:

Based on parameter tunning step we can see that

a. 5 most important features are -
['teen_humid','laundry_temp','station_humid','kitchen_humid','laundry_humid']

b. 5 least important features are -
['bath_temp','ironing_temp','kitchen_temp','parents_temp','Visibility']

As can be observed with R2 Score , compared to Tuned model 0.63 the R2 score has come down to 0.50 which is increase of 21% . The reduction in R2 score is high and we should not use reduced feature set for this data set

```
Training set R2 Score -  0.9999837366892088
Testing set R2 Score -  0.4959970347356365
Testing set RMSE Score -  0.6987523962686669
```

Feature Engineering

- Column Management.

```
# Divide the columns based on type for clear column management

col_temp = ['kitchen_temp', 'living_temp', 'laundry_temp', 'office_temp',
            'bath_temp', 'outside_temp', 'ironing_temp', 'teen_temp', 'parents_temp']

col_hum = ['kitchen_humid', 'living_humid', 'laundry_humid', 'office_humid',
           'bath_humid', 'outside_humid', 'ironing_humid', 'teen_humid', 'parents_humid']

col_weather = ['station_temp', 'Tdewpoint','station_humid',"Press_mm_hg",
               "Windspeed","Visibility"]

col_randoms = ["rv1", "rv2"]

col_target = ["Appliances"]
```

- Working on Multi-Collinearity.

- Detecting Multi-Collinearity using VIF.

Feature Engineering

	variables	VIF
0	kitchen_temp	3603.955627
1	kitchen_humid	1638.935566
2	living_temp	2490.023473
3	living_humid	2163.849100
4	laundry_temp	1239.191588
5	laundry_humid	1567.810560
6	office_temp	932.769026
7	office_humid	1357.805623
8	bath_temp	1187.570988
9	bath_humid	45.083892
10	outside_temp	88.922673
11	outside_humid	40.320200
12	ironing_temp	1613.397546
13	ironing_humid	518.841853
14	teen_temp	975.048702
15	teen_humid	568.398725
16	parents_temp	2517.024162
17	parents_humid	637.364594
18	station_temp	399.713776
19	Press_mm_hg	2084.651179
20	station_humid	1297.946141
21	Windspeed	5.245759
22	Visibility	12.029581
23	Tdewpoint	132.477783
24	rv1	inf
25	rv2	inf

Let us write a function for calculating heat index.

$$\mathrm{HI} = c_1 + c_2T + c_3R + c_4TR + c_5T^2 + c_6R^2 + c_7T^2R + c_8TR^2 + c_9T^2R^2$$

The following coefficients can be used to determine the heat index when the temperature is given in degrees Celsius, where:
HI = Heat Index (in degrees Celsius)
T = Temperature (in degrees Celsius)
R = Humidity (in %)
c1 = -8.78469475556
c2 = 1.61139411
c3 = 2.33854883889
c4 = -0.14611605
c5 = -0.012308094
c6 = -0.0164248277778
c7 = 0.002211732
c8 = 0.00072546
c9 = -0.000003582

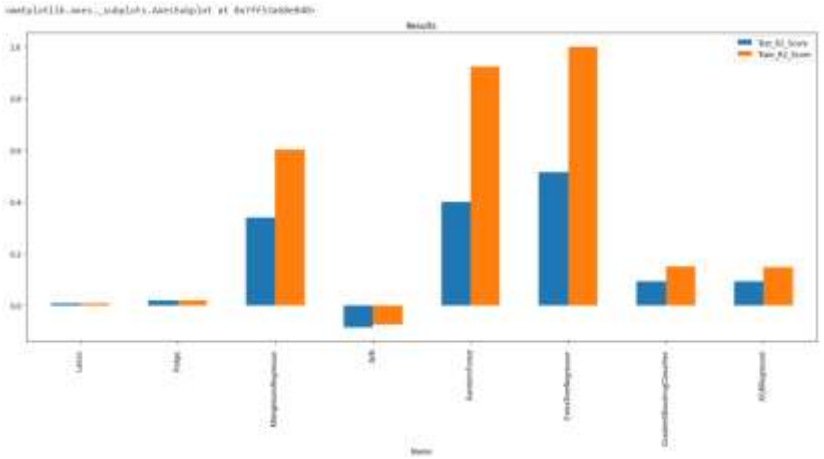
	variables	VIF
0	kitchen_feel	37859.372204
1	living_feel	4237.617302
2	laundry_feel	9045.650742
3	office_feel	11012.249304
4	bath_feel	607.985530
5	outside_feel	38.720876
6	ironing_feel	12302.412649
7	teen_feel	9297.481136
8	parents_feel	10691.358656
9	station_feel	167.717545
10	Press_mm_hg	5539.187643
11	Windspeed	4.946103
12	Visibility	11.814341
13	Tdewpoint	20.181479

	variables	VIF
0	Press_mm_hg	3304.977548
1	Windspeed	4.470945
2	Visibility	11.708678
3	Tdewpoint	5.872201
4	home_feel	3831.105539
5	out_feel	24.944033

	variables	VIF
0	Windspeed	4.176246
1	Visibility	11.690395
2	Tdewpoint	5.116533
3	home_feel	50.314480
4	out_feel	22.623470

Model Performance – After reducing Multi-Collinearity

	Name	Train_R2_Score	Test_R2_Score	Test_RMSE_Score	Train_RMSE_Score
0	Lasso:	0.008459	0.007362	105.784782	100.839731
1	Ridge:	0.020389	0.020224	105.097212	100.231251
2	KNeighborsRegressor:	0.603957	0.339739	86.275146	63.730515
3	SVR:	-0.073940	-0.083584	110.524623	104.946122
4	RandomForest	0.923077	0.400093	82.175787	28.088955
5	ExtraTreeRegressor:	1.000000	0.514148	74.008194	0.049373
6	GradientBoostingClassifier:	0.152072	0.092298	101.157818	93.251496
7	XGBRegressor:	0.148503	0.093801	101.074016	93.447545



- We can see an improvement in the Lasso regression, obvious due to removal of several features.
- Ridge underperformed by 15% when compared with its performance by using all features.
- K-Nearest underperformed by 31% when compared with its performance by using all features.
- SVR has performed the worst.
- Random Forest underperformed by 31% when compared with its performance by using all features, and the hyper-parameter tuned model.

Conclusion

- Dataset doesn’t have any null values.
- We have observed very less co-relation between the target and feature variables.
- Dropped features like rv1 & rv2 as it has infinity VIF.
- Top 2 models were Extra Tree Regressor & Random Forest.
- Worked on Multi-Collinearity, but not much significant effect on the dataset.
- Tree based models are the best ones while dealing with features which has very less correlation with the target variable. Thus, the linear models, Ridge & Lasso performed the worst.

Scope of Improvement

- We can work on the day/week feature to explore more on the model performance.
- We can try various hyper-parameter tuning methods.
- We had worked on Boruta Feature Selection for Extra Tree Regressor & Random Forest. The model had shown good results comparatively, but due to time constraint could not focus more on it.

	Name	Train_Time	Train_R2_Score	Test_R2_Score	Test_RMSE_Score
0	RandomForest	29.715513	0.939326	0.555715	0.666547
1	ExtraTreeRegressor :	7.424905	1.000000	0.632617	0.606121