Instructions:

In this Assignment, you will demonstrate the data regression skills you have learned by completing this course. You are expected to leverage a wide variety of tools, but also this report should focus on present findings, insights, and next steps. You may include some visuals from your code output, but this report is intended as a summary of your findings, not as a code review.

The grading will center around 5 main points:

1. Does the report include a section describing the data?
2. Does the report include a paragraph detailing the main objective(s) of this analysis?
3. Does the report include a section with variations of linear regression models and specifies which one is the model that best suits the main objective(s) of this analysis.
4. Does the report include a clear and well-presented section with key findings related to the main objective(s) of the analysis?
5. Does the report highlight possible flaws in the model and a plan of action to revisit this analysis with additional data or different predictive modeling techniques?

Import the required libraries

The following required modules are pre-installed in the Skills Network Labs environment. However if you run this notebook commands in a different Jupyter environment (e.g. Watson Studio or Ananconda) you will need to install these libraries by removing the `#` sign before `!mamba` in the code cell below.

```
# All Libraries required for this lab are listed below. The libraries
pre-installed on Skills Network Labs are commented.
# !mamba install -qy pandas==1.3.4 numpy==1.21.4 seaborn==0.9.0
matplotlib==3.5.0 scikit-learn==0.20.1
# Note: If your environment doesn't support "!mamba install", use "!
pip install"

# Surpress warnings:
def warn(*args, **kwargs):
    pass
import warnings
warnings.warn = warn

import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pylab as plt
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.linear_model import
LinearRegression,Ridge,Lasso,ElasticNet
```

```
from sklearn.metrics import r2_score
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import scale
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import SelectKBest, f_regression
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.decomposition import PCA
```

Importing the Dataset

Before you begin, you will need to choose a data set that you feel passionate about. You can brainstorm with your peers about great public data sets using the discussion board in this module.

Read your chosen dataset into pandas dataframe:

Dataset taken from Kaggle.

```
data = pd.read_csv('data/Energy_consumption.csv')
data.head(15)

              Timestamp  Temperature   Humidity  SquareFootage
Occupancy  \
0   2022-01-01 00:00:00    25.139433  43.431581    1565.693999
5
1   2022-01-01 01:00:00    27.731651  54.225919    1411.064918
1
2   2022-01-01 02:00:00    28.704277  58.907658    1755.715009
2
3   2022-01-01 03:00:00    20.080469  50.371637    1452.316318
1
4   2022-01-01 04:00:00    23.097359  51.401421    1094.130359
9
5   2022-01-01 05:00:00    29.576037  36.824263    1871.709180
6
6   2022-01-01 06:00:00    25.131167  35.709622    1607.001228
6
7   2022-01-01 07:00:00    23.182844  31.679920    1633.955330
8
8   2022-01-01 08:00:00    25.391999  46.399364    1240.309224
6
9   2022-01-01 09:00:00    22.212549  32.418464    1705.420336
1
10  2022-01-01 10:00:00    28.064814  36.451472    1341.467129
2
11  2022-01-01 11:00:00    23.422546  30.527342    1604.418355
```

```
6
12  2022-01-01 12:00:00    25.388888  47.601018    1244.618914
1
13  2022-01-01 13:00:00    20.058738  41.861642    1806.052632
2
14  2022-01-01 14:00:00    26.731525  37.297870    1419.749014
6

   HVACUsage LightingUsage  RenewableEnergy  DayOfWeek Holiday  \
0         On           Off         2.774699     Monday      No
1         On            On        21.831384   Saturday      No
2        Off           Off         6.764672     Sunday      No
3        Off            On         8.623447  Wednesday      No
4         On           Off         3.071969     Friday      No
5        Off           Off        17.626690     Sunday     Yes
6         On           Off        24.264702     Friday     Yes
7        Off           Off        27.517099   Thursday     Yes
8         On           Off         2.307595     Sunday      No
9         On           Off        29.140071    Tuesday      No
10       Off           Off         0.352238     Monday     Yes
11        On            On        19.529548   Thursday     Yes
12        On           Off        21.797444    Tuesday     Yes
13       Off           Off         6.384949     Friday     Yes
14       Off           Off        12.074223     Friday     Yes

    EnergyConsumption
0           75.364373
1           83.401855
2           78.270888
3           56.519850
4           70.811732
5           84.321885
6           76.165791
7           74.131906
8           78.206236
9           77.992214
10          82.274434
11          73.278670
12          84.144776
13          60.022519
14          81.183188
```

Once you have selected a data set, you will produce the deliverables listed below and submit them to one of your peers for review. Treat this exercise as an opportunity to produce analysis that are ready to highlight your analytical skills for a senior audience, for example, the Chief Data Officer, or the Head of Analytics at your company. Sections required in your report:

- Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation.
- Brief description of the data set you chose and a summary of its attributes.

- Brief summary of data exploration and actions taken for data cleaning and feature engineering.
- Summary of training at least three linear regression models which should be variations that cover using a simple linear regression as a baseline, adding polynomial effects, and using a regularization regression. Preferably, all use the same training and test splits, or the same cross-validation method.
- A paragraph explaining which of your regressions you recommend as a final model that best fits your needs in terms of accuracy and explainability.
- Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your linear regression model.
- Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model adding specific data features to achieve a better explanation or a better prediction.

1. About the Data

This dataset encapsulates a diverse array of features, including temperature, humidity, occupancy, HVAC and lighting usage, renewable energy contributions, and more. Each timestamp provides a snapshot of a hypothetical environment, allowing for in-depth analysis and modeling of energy consumption behaviors. Dive into the nuances of this synthetic dataset, designed to emulate real-world scenarios, and unravel the complexities that influence energy usage. Whether you are delving into predictive modeling or honing your data analysis skills, this dataset offers a dynamic playground for experimentation and discovery.

```
data.columns

Index(['Timestamp', 'Temperature', 'Humidity', 'SquareFootage',
'Occupancy',
       'HVACUsage', 'LightingUsage', 'RenewableEnergy', 'DayOfWeek',
'Holiday',
       'EnergyConsumption'],
      dtype='object')

sum(data.duplicated())

0

data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Timestamp          1000 non-null   object
 1   Temperature        1000 non-null   float64
 2   Humidity           1000 non-null   float64
 3   SquareFootage      1000 non-null   float64
```

```
 4   Occupancy          1000 non-null   int64
 5   HVACUsage          1000 non-null   object
 6   LightingUsage      1000 non-null   object
 7   RenewableEnergy    1000 non-null   float64
 8   DayOfWeek          1000 non-null   object
 9   Holiday            1000 non-null   object
 10  EnergyConsumption  1000 non-null   float64
dtypes: float64(5), int64(1), object(5)
memory usage: 86.1+ KB
```

```
pd.DataFrame(data.isnull().value_counts()).T
```

```
Timestamp         False
Temperature       False
Humidity          False
SquareFootage     False
Occupancy         False
HVACUsage         False
LightingUsage     False
RenewableEnergy   False
DayOfWeek         False
Holiday           False
EnergyConsumption False
count              1000
```

```
data.describe().T
```

```
                    count         mean         std          min
25%  \
Temperature        1000.0    24.982026    2.836850    20.007565
22.645070
Humidity           1000.0    45.395412    8.518905    30.015975
38.297722
SquareFootage      1000.0  1500.052488  288.418873  1000.512661
1247.108548
Occupancy          1000.0     4.581000    2.865598     0.000000
2.000000
RenewableEnergy    1000.0    15.132813    8.745917     0.006642
7.628385
EnergyConsumption  1000.0    77.055873    8.144112    53.263278
71.544690

                           50%          75%          max
Temperature          24.751637    27.418174    29.998671
Humidity             45.972116    52.420066    59.969085
SquareFootage      1507.967426  1740.340165  1999.982252
Occupancy             5.000000     7.000000     9.000000
RenewableEnergy      15.072296    22.884064    29.965327
EnergyConsumption    76.943696    82.921742    99.201120
```

```python
feature_cols = data.select_dtypes(include=object)
for col in feature_cols.columns:
    print(col,':',data[col].unique(),'\n')
```

```
Timestamp : ['2022-01-01 00:00:00' '2022-01-01 01:00:00' '2022-01-01
02:00:00'
 '2022-01-01 03:00:00' '2022-01-01 04:00:00' '2022-01-01 05:00:00'
 '2022-01-01 06:00:00' '2022-01-01 07:00:00' '2022-01-01 08:00:00'
 '2022-01-01 09:00:00' '2022-01-01 10:00:00' '2022-01-01 11:00:00'
 '2022-01-01 12:00:00' '2022-01-01 13:00:00' '2022-01-01 14:00:00'
 '2022-01-01 15:00:00' '2022-01-01 16:00:00' '2022-01-01 17:00:00'
 '2022-01-01 18:00:00' '2022-01-01 19:00:00' '2022-01-01 20:00:00'
 '2022-01-01 21:00:00' '2022-01-01 22:00:00' '2022-01-01 23:00:00'
 '2022-01-02 00:00:00' '2022-01-02 01:00:00' '2022-01-02 02:00:00'
 '2022-01-02 03:00:00' '2022-01-02 04:00:00' '2022-01-02 05:00:00'
 '2022-01-02 06:00:00' '2022-01-02 07:00:00' '2022-01-02 08:00:00'
 '2022-01-02 09:00:00' '2022-01-02 10:00:00' '2022-01-02 11:00:00'
 '2022-01-02 12:00:00' '2022-01-02 13:00:00' '2022-01-02 14:00:00'
 '2022-01-02 15:00:00' '2022-01-02 16:00:00' '2022-01-02 17:00:00'
 '2022-01-02 18:00:00' '2022-01-02 19:00:00' '2022-01-02 20:00:00'
 '2022-01-02 21:00:00' '2022-01-02 22:00:00' '2022-01-02 23:00:00'
 '2022-01-03 00:00:00' '2022-01-03 01:00:00' '2022-01-03 02:00:00'
 '2022-01-03 03:00:00' '2022-01-03 04:00:00' '2022-01-03 05:00:00'
 '2022-01-03 06:00:00' '2022-01-03 07:00:00' '2022-01-03 08:00:00'
 '2022-01-03 09:00:00' '2022-01-03 10:00:00' '2022-01-03 11:00:00'
 '2022-01-03 12:00:00' '2022-01-03 13:00:00' '2022-01-03 14:00:00'
 '2022-01-03 15:00:00' '2022-01-03 16:00:00' '2022-01-03 17:00:00'
 '2022-01-03 18:00:00' '2022-01-03 19:00:00' '2022-01-03 20:00:00'
 '2022-01-03 21:00:00' '2022-01-03 22:00:00' '2022-01-03 23:00:00'
 '2022-01-04 00:00:00' '2022-01-04 01:00:00' '2022-01-04 02:00:00'
 '2022-01-04 03:00:00' '2022-01-04 04:00:00' '2022-01-04 05:00:00'
 '2022-01-04 06:00:00' '2022-01-04 07:00:00' '2022-01-04 08:00:00'
 '2022-01-04 09:00:00' '2022-01-04 10:00:00' '2022-01-04 11:00:00'
 '2022-01-04 12:00:00' '2022-01-04 13:00:00' '2022-01-04 14:00:00'
 '2022-01-04 15:00:00' '2022-01-04 16:00:00' '2022-01-04 17:00:00'
 '2022-01-04 18:00:00' '2022-01-04 19:00:00' '2022-01-04 20:00:00'
 '2022-01-04 21:00:00' '2022-01-04 22:00:00' '2022-01-04 23:00:00'
 '2022-01-05 00:00:00' '2022-01-05 01:00:00' '2022-01-05 02:00:00'
 '2022-01-05 03:00:00' '2022-01-05 04:00:00' '2022-01-05 05:00:00'
 '2022-01-05 06:00:00' '2022-01-05 07:00:00' '2022-01-05 08:00:00'
 '2022-01-05 09:00:00' '2022-01-05 10:00:00' '2022-01-05 11:00:00'
 '2022-01-05 12:00:00' '2022-01-05 13:00:00' '2022-01-05 14:00:00'
 '2022-01-05 15:00:00' '2022-01-05 16:00:00' '2022-01-05 17:00:00'
 '2022-01-05 18:00:00' '2022-01-05 19:00:00' '2022-01-05 20:00:00'
 '2022-01-05 21:00:00' '2022-01-05 22:00:00' '2022-01-05 23:00:00'
 '2022-01-06 00:00:00' '2022-01-06 01:00:00' '2022-01-06 02:00:00'
 '2022-01-06 03:00:00' '2022-01-06 04:00:00' '2022-01-06 05:00:00'
 '2022-01-06 06:00:00' '2022-01-06 07:00:00' '2022-01-06 08:00:00'
 '2022-01-06 09:00:00' '2022-01-06 10:00:00' '2022-01-06 11:00:00'
 '2022-01-06 12:00:00' '2022-01-06 13:00:00' '2022-01-06 14:00:00'
```

```
 '2022-01-06 15:00:00' '2022-01-06 16:00:00' '2022-01-06 17:00:00'
 '2022-01-06 18:00:00' '2022-01-06 19:00:00' '2022-01-06 20:00:00'
 '2022-01-06 21:00:00' '2022-01-06 22:00:00' '2022-01-06 23:00:00'
 '2022-01-07 00:00:00' '2022-01-07 01:00:00' '2022-01-07 02:00:00'
 '2022-01-07 03:00:00' '2022-01-07 04:00:00' '2022-01-07 05:00:00'
 '2022-01-07 06:00:00' '2022-01-07 07:00:00' '2022-01-07 08:00:00'
 '2022-01-07 09:00:00' '2022-01-07 10:00:00' '2022-01-07 11:00:00'
 '2022-01-07 12:00:00' '2022-01-07 13:00:00' '2022-01-07 14:00:00'
 '2022-01-07 15:00:00' '2022-01-07 16:00:00' '2022-01-07 17:00:00'
 '2022-01-07 18:00:00' '2022-01-07 19:00:00' '2022-01-07 20:00:00'
 '2022-01-07 21:00:00' '2022-01-07 22:00:00' '2022-01-07 23:00:00'
 '2022-01-08 00:00:00' '2022-01-08 01:00:00' '2022-01-08 02:00:00'
 '2022-01-08 03:00:00' '2022-01-08 04:00:00' '2022-01-08 05:00:00'
 '2022-01-08 06:00:00' '2022-01-08 07:00:00' '2022-01-08 08:00:00'
 '2022-01-08 09:00:00' '2022-01-08 10:00:00' '2022-01-08 11:00:00'
 '2022-01-08 12:00:00' '2022-01-08 13:00:00' '2022-01-08 14:00:00'
 '2022-01-08 15:00:00' '2022-01-08 16:00:00' '2022-01-08 17:00:00'
 '2022-01-08 18:00:00' '2022-01-08 19:00:00' '2022-01-08 20:00:00'
 '2022-01-08 21:00:00' '2022-01-08 22:00:00' '2022-01-08 23:00:00'
 '2022-01-09 00:00:00' '2022-01-09 01:00:00' '2022-01-09 02:00:00'
 '2022-01-09 03:00:00' '2022-01-09 04:00:00' '2022-01-09 05:00:00'
 '2022-01-09 06:00:00' '2022-01-09 07:00:00' '2022-01-09 08:00:00'
 '2022-01-09 09:00:00' '2022-01-09 10:00:00' '2022-01-09 11:00:00'
 '2022-01-09 12:00:00' '2022-01-09 13:00:00' '2022-01-09 14:00:00'
 '2022-01-09 15:00:00' '2022-01-09 16:00:00' '2022-01-09 17:00:00'
 '2022-01-09 18:00:00' '2022-01-09 19:00:00' '2022-01-09 20:00:00'
 '2022-01-09 21:00:00' '2022-01-09 22:00:00' '2022-01-09 23:00:00'
 '2022-01-10 00:00:00' '2022-01-10 01:00:00' '2022-01-10 02:00:00'
 '2022-01-10 03:00:00' '2022-01-10 04:00:00' '2022-01-10 05:00:00'
 '2022-01-10 06:00:00' '2022-01-10 07:00:00' '2022-01-10 08:00:00'
 '2022-01-10 09:00:00' '2022-01-10 10:00:00' '2022-01-10 11:00:00'
 '2022-01-10 12:00:00' '2022-01-10 13:00:00' '2022-01-10 14:00:00'
 '2022-01-10 15:00:00' '2022-01-10 16:00:00' '2022-01-10 17:00:00'
 '2022-01-10 18:00:00' '2022-01-10 19:00:00' '2022-01-10 20:00:00'
 '2022-01-10 21:00:00' '2022-01-10 22:00:00' '2022-01-10 23:00:00'
 '2022-01-11 00:00:00' '2022-01-11 01:00:00' '2022-01-11 02:00:00'
 '2022-01-11 03:00:00' '2022-01-11 04:00:00' '2022-01-11 05:00:00'
 '2022-01-11 06:00:00' '2022-01-11 07:00:00' '2022-01-11 08:00:00'
 '2022-01-11 09:00:00' '2022-01-11 10:00:00' '2022-01-11 11:00:00'
 '2022-01-11 12:00:00' '2022-01-11 13:00:00' '2022-01-11 14:00:00'
 '2022-01-11 15:00:00' '2022-01-11 16:00:00' '2022-01-11 17:00:00'
 '2022-01-11 18:00:00' '2022-01-11 19:00:00' '2022-01-11 20:00:00'
 '2022-01-11 21:00:00' '2022-01-11 22:00:00' '2022-01-11 23:00:00'
 '2022-01-12 00:00:00' '2022-01-12 01:00:00' '2022-01-12 02:00:00'
 '2022-01-12 03:00:00' '2022-01-12 04:00:00' '2022-01-12 05:00:00'
 '2022-01-12 06:00:00' '2022-01-12 07:00:00' '2022-01-12 08:00:00'
 '2022-01-12 09:00:00' '2022-01-12 10:00:00' '2022-01-12 11:00:00'
 '2022-01-12 12:00:00' '2022-01-12 13:00:00' '2022-01-12 14:00:00'
 '2022-01-12 15:00:00' '2022-01-12 16:00:00' '2022-01-12 17:00:00'
```

```
 '2022-01-12 18:00:00' '2022-01-12 19:00:00' '2022-01-12 20:00:00'
 '2022-01-12 21:00:00' '2022-01-12 22:00:00' '2022-01-12 23:00:00'
 '2022-01-13 00:00:00' '2022-01-13 01:00:00' '2022-01-13 02:00:00'
 '2022-01-13 03:00:00' '2022-01-13 04:00:00' '2022-01-13 05:00:00'
 '2022-01-13 06:00:00' '2022-01-13 07:00:00' '2022-01-13 08:00:00'
 '2022-01-13 09:00:00' '2022-01-13 10:00:00' '2022-01-13 11:00:00'
 '2022-01-13 12:00:00' '2022-01-13 13:00:00' '2022-01-13 14:00:00'
 '2022-01-13 15:00:00' '2022-01-13 16:00:00' '2022-01-13 17:00:00'
 '2022-01-13 18:00:00' '2022-01-13 19:00:00' '2022-01-13 20:00:00'
 '2022-01-13 21:00:00' '2022-01-13 22:00:00' '2022-01-13 23:00:00'
 '2022-01-14 00:00:00' '2022-01-14 01:00:00' '2022-01-14 02:00:00'
 '2022-01-14 03:00:00' '2022-01-14 04:00:00' '2022-01-14 05:00:00'
 '2022-01-14 06:00:00' '2022-01-14 07:00:00' '2022-01-14 08:00:00'
 '2022-01-14 09:00:00' '2022-01-14 10:00:00' '2022-01-14 11:00:00'
 '2022-01-14 12:00:00' '2022-01-14 13:00:00' '2022-01-14 14:00:00'
 '2022-01-14 15:00:00' '2022-01-14 16:00:00' '2022-01-14 17:00:00'
 '2022-01-14 18:00:00' '2022-01-14 19:00:00' '2022-01-14 20:00:00'
 '2022-01-14 21:00:00' '2022-01-14 22:00:00' '2022-01-14 23:00:00'
 '2022-01-15 00:00:00' '2022-01-15 01:00:00' '2022-01-15 02:00:00'
 '2022-01-15 03:00:00' '2022-01-15 04:00:00' '2022-01-15 05:00:00'
 '2022-01-15 06:00:00' '2022-01-15 07:00:00' '2022-01-15 08:00:00'
 '2022-01-15 09:00:00' '2022-01-15 10:00:00' '2022-01-15 11:00:00'
 '2022-01-15 12:00:00' '2022-01-15 13:00:00' '2022-01-15 14:00:00'
 '2022-01-15 15:00:00' '2022-01-15 16:00:00' '2022-01-15 17:00:00'
 '2022-01-15 18:00:00' '2022-01-15 19:00:00' '2022-01-15 20:00:00'
 '2022-01-15 21:00:00' '2022-01-15 22:00:00' '2022-01-15 23:00:00'
 '2022-01-16 00:00:00' '2022-01-16 01:00:00' '2022-01-16 02:00:00'
 '2022-01-16 03:00:00' '2022-01-16 04:00:00' '2022-01-16 05:00:00'
 '2022-01-16 06:00:00' '2022-01-16 07:00:00' '2022-01-16 08:00:00'
 '2022-01-16 09:00:00' '2022-01-16 10:00:00' '2022-01-16 11:00:00'
 '2022-01-16 12:00:00' '2022-01-16 13:00:00' '2022-01-16 14:00:00'
 '2022-01-16 15:00:00' '2022-01-16 16:00:00' '2022-01-16 17:00:00'
 '2022-01-16 18:00:00' '2022-01-16 19:00:00' '2022-01-16 20:00:00'
 '2022-01-16 21:00:00' '2022-01-16 22:00:00' '2022-01-16 23:00:00'
 '2022-01-17 00:00:00' '2022-01-17 01:00:00' '2022-01-17 02:00:00'
 '2022-01-17 03:00:00' '2022-01-17 04:00:00' '2022-01-17 05:00:00'
 '2022-01-17 06:00:00' '2022-01-17 07:00:00' '2022-01-17 08:00:00'
 '2022-01-17 09:00:00' '2022-01-17 10:00:00' '2022-01-17 11:00:00'
 '2022-01-17 12:00:00' '2022-01-17 13:00:00' '2022-01-17 14:00:00'
 '2022-01-17 15:00:00' '2022-01-17 16:00:00' '2022-01-17 17:00:00'
 '2022-01-17 18:00:00' '2022-01-17 19:00:00' '2022-01-17 20:00:00'
 '2022-01-17 21:00:00' '2022-01-17 22:00:00' '2022-01-17 23:00:00'
 '2022-01-18 00:00:00' '2022-01-18 01:00:00' '2022-01-18 02:00:00'
 '2022-01-18 03:00:00' '2022-01-18 04:00:00' '2022-01-18 05:00:00'
 '2022-01-18 06:00:00' '2022-01-18 07:00:00' '2022-01-18 08:00:00'
 '2022-01-18 09:00:00' '2022-01-18 10:00:00' '2022-01-18 11:00:00'
 '2022-01-18 12:00:00' '2022-01-18 13:00:00' '2022-01-18 14:00:00'
 '2022-01-18 15:00:00' '2022-01-18 16:00:00' '2022-01-18 17:00:00'
 '2022-01-18 18:00:00' '2022-01-18 19:00:00' '2022-01-18 20:00:00'
```

```
 '2022-01-18 21:00:00' '2022-01-18 22:00:00' '2022-01-18 23:00:00'
 '2022-01-19 00:00:00' '2022-01-19 01:00:00' '2022-01-19 02:00:00'
 '2022-01-19 03:00:00' '2022-01-19 04:00:00' '2022-01-19 05:00:00'
 '2022-01-19 06:00:00' '2022-01-19 07:00:00' '2022-01-19 08:00:00'
 '2022-01-19 09:00:00' '2022-01-19 10:00:00' '2022-01-19 11:00:00'
 '2022-01-19 12:00:00' '2022-01-19 13:00:00' '2022-01-19 14:00:00'
 '2022-01-19 15:00:00' '2022-01-19 16:00:00' '2022-01-19 17:00:00'
 '2022-01-19 18:00:00' '2022-01-19 19:00:00' '2022-01-19 20:00:00'
 '2022-01-19 21:00:00' '2022-01-19 22:00:00' '2022-01-19 23:00:00'
 '2022-01-20 00:00:00' '2022-01-20 01:00:00' '2022-01-20 02:00:00'
 '2022-01-20 03:00:00' '2022-01-20 04:00:00' '2022-01-20 05:00:00'
 '2022-01-20 06:00:00' '2022-01-20 07:00:00' '2022-01-20 08:00:00'
 '2022-01-20 09:00:00' '2022-01-20 10:00:00' '2022-01-20 11:00:00'
 '2022-01-20 12:00:00' '2022-01-20 13:00:00' '2022-01-20 14:00:00'
 '2022-01-20 15:00:00' '2022-01-20 16:00:00' '2022-01-20 17:00:00'
 '2022-01-20 18:00:00' '2022-01-20 19:00:00' '2022-01-20 20:00:00'
 '2022-01-20 21:00:00' '2022-01-20 22:00:00' '2022-01-20 23:00:00'
 '2022-01-21 00:00:00' '2022-01-21 01:00:00' '2022-01-21 02:00:00'
 '2022-01-21 03:00:00' '2022-01-21 04:00:00' '2022-01-21 05:00:00'
 '2022-01-21 06:00:00' '2022-01-21 07:00:00' '2022-01-21 08:00:00'
 '2022-01-21 09:00:00' '2022-01-21 10:00:00' '2022-01-21 11:00:00'
 '2022-01-21 12:00:00' '2022-01-21 13:00:00' '2022-01-21 14:00:00'
 '2022-01-21 15:00:00' '2022-01-21 16:00:00' '2022-01-21 17:00:00'
 '2022-01-21 18:00:00' '2022-01-21 19:00:00' '2022-01-21 20:00:00'
 '2022-01-21 21:00:00' '2022-01-21 22:00:00' '2022-01-21 23:00:00'
 '2022-01-22 00:00:00' '2022-01-22 01:00:00' '2022-01-22 02:00:00'
 '2022-01-22 03:00:00' '2022-01-22 04:00:00' '2022-01-22 05:00:00'
 '2022-01-22 06:00:00' '2022-01-22 07:00:00' '2022-01-22 08:00:00'
 '2022-01-22 09:00:00' '2022-01-22 10:00:00' '2022-01-22 11:00:00'
 '2022-01-22 12:00:00' '2022-01-22 13:00:00' '2022-01-22 14:00:00'
 '2022-01-22 15:00:00' '2022-01-22 16:00:00' '2022-01-22 17:00:00'
 '2022-01-22 18:00:00' '2022-01-22 19:00:00' '2022-01-22 20:00:00'
 '2022-01-22 21:00:00' '2022-01-22 22:00:00' '2022-01-22 23:00:00'
 '2022-01-23 00:00:00' '2022-01-23 01:00:00' '2022-01-23 02:00:00'
 '2022-01-23 03:00:00' '2022-01-23 04:00:00' '2022-01-23 05:00:00'
 '2022-01-23 06:00:00' '2022-01-23 07:00:00' '2022-01-23 08:00:00'
 '2022-01-23 09:00:00' '2022-01-23 10:00:00' '2022-01-23 11:00:00'
 '2022-01-23 12:00:00' '2022-01-23 13:00:00' '2022-01-23 14:00:00'
 '2022-01-23 15:00:00' '2022-01-23 16:00:00' '2022-01-23 17:00:00'
 '2022-01-23 18:00:00' '2022-01-23 19:00:00' '2022-01-23 20:00:00'
 '2022-01-23 21:00:00' '2022-01-23 22:00:00' '2022-01-23 23:00:00'
 '2022-01-24 00:00:00' '2022-01-24 01:00:00' '2022-01-24 02:00:00'
 '2022-01-24 03:00:00' '2022-01-24 04:00:00' '2022-01-24 05:00:00'
 '2022-01-24 06:00:00' '2022-01-24 07:00:00' '2022-01-24 08:00:00'
 '2022-01-24 09:00:00' '2022-01-24 10:00:00' '2022-01-24 11:00:00'
 '2022-01-24 12:00:00' '2022-01-24 13:00:00' '2022-01-24 14:00:00'
 '2022-01-24 15:00:00' '2022-01-24 16:00:00' '2022-01-24 17:00:00'
 '2022-01-24 18:00:00' '2022-01-24 19:00:00' '2022-01-24 20:00:00'
 '2022-01-24 21:00:00' '2022-01-24 22:00:00' '2022-01-24 23:00:00'
```

```
 '2022-01-25 00:00:00' '2022-01-25 01:00:00' '2022-01-25 02:00:00'
 '2022-01-25 03:00:00' '2022-01-25 04:00:00' '2022-01-25 05:00:00'
 '2022-01-25 06:00:00' '2022-01-25 07:00:00' '2022-01-25 08:00:00'
 '2022-01-25 09:00:00' '2022-01-25 10:00:00' '2022-01-25 11:00:00'
 '2022-01-25 12:00:00' '2022-01-25 13:00:00' '2022-01-25 14:00:00'
 '2022-01-25 15:00:00' '2022-01-25 16:00:00' '2022-01-25 17:00:00'
 '2022-01-25 18:00:00' '2022-01-25 19:00:00' '2022-01-25 20:00:00'
 '2022-01-25 21:00:00' '2022-01-25 22:00:00' '2022-01-25 23:00:00'
 '2022-01-26 00:00:00' '2022-01-26 01:00:00' '2022-01-26 02:00:00'
 '2022-01-26 03:00:00' '2022-01-26 04:00:00' '2022-01-26 05:00:00'
 '2022-01-26 06:00:00' '2022-01-26 07:00:00' '2022-01-26 08:00:00'
 '2022-01-26 09:00:00' '2022-01-26 10:00:00' '2022-01-26 11:00:00'
 '2022-01-26 12:00:00' '2022-01-26 13:00:00' '2022-01-26 14:00:00'
 '2022-01-26 15:00:00' '2022-01-26 16:00:00' '2022-01-26 17:00:00'
 '2022-01-26 18:00:00' '2022-01-26 19:00:00' '2022-01-26 20:00:00'
 '2022-01-26 21:00:00' '2022-01-26 22:00:00' '2022-01-26 23:00:00'
 '2022-01-27 00:00:00' '2022-01-27 01:00:00' '2022-01-27 02:00:00'
 '2022-01-27 03:00:00' '2022-01-27 04:00:00' '2022-01-27 05:00:00'
 '2022-01-27 06:00:00' '2022-01-27 07:00:00' '2022-01-27 08:00:00'
 '2022-01-27 09:00:00' '2022-01-27 10:00:00' '2022-01-27 11:00:00'
 '2022-01-27 12:00:00' '2022-01-27 13:00:00' '2022-01-27 14:00:00'
 '2022-01-27 15:00:00' '2022-01-27 16:00:00' '2022-01-27 17:00:00'
 '2022-01-27 18:00:00' '2022-01-27 19:00:00' '2022-01-27 20:00:00'
 '2022-01-27 21:00:00' '2022-01-27 22:00:00' '2022-01-27 23:00:00'
 '2022-01-28 00:00:00' '2022-01-28 01:00:00' '2022-01-28 02:00:00'
 '2022-01-28 03:00:00' '2022-01-28 04:00:00' '2022-01-28 05:00:00'
 '2022-01-28 06:00:00' '2022-01-28 07:00:00' '2022-01-28 08:00:00'
 '2022-01-28 09:00:00' '2022-01-28 10:00:00' '2022-01-28 11:00:00'
 '2022-01-28 12:00:00' '2022-01-28 13:00:00' '2022-01-28 14:00:00'
 '2022-01-28 15:00:00' '2022-01-28 16:00:00' '2022-01-28 17:00:00'
 '2022-01-28 18:00:00' '2022-01-28 19:00:00' '2022-01-28 20:00:00'
 '2022-01-28 21:00:00' '2022-01-28 22:00:00' '2022-01-28 23:00:00'
 '2022-01-29 00:00:00' '2022-01-29 01:00:00' '2022-01-29 02:00:00'
 '2022-01-29 03:00:00' '2022-01-29 04:00:00' '2022-01-29 05:00:00'
 '2022-01-29 06:00:00' '2022-01-29 07:00:00' '2022-01-29 08:00:00'
 '2022-01-29 09:00:00' '2022-01-29 10:00:00' '2022-01-29 11:00:00'
 '2022-01-29 12:00:00' '2022-01-29 13:00:00' '2022-01-29 14:00:00'
 '2022-01-29 15:00:00' '2022-01-29 16:00:00' '2022-01-29 17:00:00'
 '2022-01-29 18:00:00' '2022-01-29 19:00:00' '2022-01-29 20:00:00'
 '2022-01-29 21:00:00' '2022-01-29 22:00:00' '2022-01-29 23:00:00'
 '2022-01-30 00:00:00' '2022-01-30 01:00:00' '2022-01-30 02:00:00'
 '2022-01-30 03:00:00' '2022-01-30 04:00:00' '2022-01-30 05:00:00'
 '2022-01-30 06:00:00' '2022-01-30 07:00:00' '2022-01-30 08:00:00'
 '2022-01-30 09:00:00' '2022-01-30 10:00:00' '2022-01-30 11:00:00'
 '2022-01-30 12:00:00' '2022-01-30 13:00:00' '2022-01-30 14:00:00'
 '2022-01-30 15:00:00' '2022-01-30 16:00:00' '2022-01-30 17:00:00'
 '2022-01-30 18:00:00' '2022-01-30 19:00:00' '2022-01-30 20:00:00'
 '2022-01-30 21:00:00' '2022-01-30 22:00:00' '2022-01-30 23:00:00'
 '2022-01-31 00:00:00' '2022-01-31 01:00:00' '2022-01-31 02:00:00'
```

```
 '2022-01-31 03:00:00' '2022-01-31 04:00:00' '2022-01-31 05:00:00'
 '2022-01-31 06:00:00' '2022-01-31 07:00:00' '2022-01-31 08:00:00'
 '2022-01-31 09:00:00' '2022-01-31 10:00:00' '2022-01-31 11:00:00'
 '2022-01-31 12:00:00' '2022-01-31 13:00:00' '2022-01-31 14:00:00'
 '2022-01-31 15:00:00' '2022-01-31 16:00:00' '2022-01-31 17:00:00'
 '2022-01-31 18:00:00' '2022-01-31 19:00:00' '2022-01-31 20:00:00'
 '2022-01-31 21:00:00' '2022-01-31 22:00:00' '2022-01-31 23:00:00'
 '2022-02-01 00:00:00' '2022-02-01 01:00:00' '2022-02-01 02:00:00'
 '2022-02-01 03:00:00' '2022-02-01 04:00:00' '2022-02-01 05:00:00'
 '2022-02-01 06:00:00' '2022-02-01 07:00:00' '2022-02-01 08:00:00'
 '2022-02-01 09:00:00' '2022-02-01 10:00:00' '2022-02-01 11:00:00'
 '2022-02-01 12:00:00' '2022-02-01 13:00:00' '2022-02-01 14:00:00'
 '2022-02-01 15:00:00' '2022-02-01 16:00:00' '2022-02-01 17:00:00'
 '2022-02-01 18:00:00' '2022-02-01 19:00:00' '2022-02-01 20:00:00'
 '2022-02-01 21:00:00' '2022-02-01 22:00:00' '2022-02-01 23:00:00'
 '2022-02-02 00:00:00' '2022-02-02 01:00:00' '2022-02-02 02:00:00'
 '2022-02-02 03:00:00' '2022-02-02 04:00:00' '2022-02-02 05:00:00'
 '2022-02-02 06:00:00' '2022-02-02 07:00:00' '2022-02-02 08:00:00'
 '2022-02-02 09:00:00' '2022-02-02 10:00:00' '2022-02-02 11:00:00'
 '2022-02-02 12:00:00' '2022-02-02 13:00:00' '2022-02-02 14:00:00'
 '2022-02-02 15:00:00' '2022-02-02 16:00:00' '2022-02-02 17:00:00'
 '2022-02-02 18:00:00' '2022-02-02 19:00:00' '2022-02-02 20:00:00'
 '2022-02-02 21:00:00' '2022-02-02 22:00:00' '2022-02-02 23:00:00'
 '2022-02-03 00:00:00' '2022-02-03 01:00:00' '2022-02-03 02:00:00'
 '2022-02-03 03:00:00' '2022-02-03 04:00:00' '2022-02-03 05:00:00'
 '2022-02-03 06:00:00' '2022-02-03 07:00:00' '2022-02-03 08:00:00'
 '2022-02-03 09:00:00' '2022-02-03 10:00:00' '2022-02-03 11:00:00'
 '2022-02-03 12:00:00' '2022-02-03 13:00:00' '2022-02-03 14:00:00'
 '2022-02-03 15:00:00' '2022-02-03 16:00:00' '2022-02-03 17:00:00'
 '2022-02-03 18:00:00' '2022-02-03 19:00:00' '2022-02-03 20:00:00'
 '2022-02-03 21:00:00' '2022-02-03 22:00:00' '2022-02-03 23:00:00'
 '2022-02-04 00:00:00' '2022-02-04 01:00:00' '2022-02-04 02:00:00'
 '2022-02-04 03:00:00' '2022-02-04 04:00:00' '2022-02-04 05:00:00'
 '2022-02-04 06:00:00' '2022-02-04 07:00:00' '2022-02-04 08:00:00'
 '2022-02-04 09:00:00' '2022-02-04 10:00:00' '2022-02-04 11:00:00'
 '2022-02-04 12:00:00' '2022-02-04 13:00:00' '2022-02-04 14:00:00'
 '2022-02-04 15:00:00' '2022-02-04 16:00:00' '2022-02-04 17:00:00'
 '2022-02-04 18:00:00' '2022-02-04 19:00:00' '2022-02-04 20:00:00'
 '2022-02-04 21:00:00' '2022-02-04 22:00:00' '2022-02-04 23:00:00'
 '2022-02-05 00:00:00' '2022-02-05 01:00:00' '2022-02-05 02:00:00'
 '2022-02-05 03:00:00' '2022-02-05 04:00:00' '2022-02-05 05:00:00'
 '2022-02-05 06:00:00' '2022-02-05 07:00:00' '2022-02-05 08:00:00'
 '2022-02-05 09:00:00' '2022-02-05 10:00:00' '2022-02-05 11:00:00'
 '2022-02-05 12:00:00' '2022-02-05 13:00:00' '2022-02-05 14:00:00'
 '2022-02-05 15:00:00' '2022-02-05 16:00:00' '2022-02-05 17:00:00'
 '2022-02-05 18:00:00' '2022-02-05 19:00:00' '2022-02-05 20:00:00'
 '2022-02-05 21:00:00' '2022-02-05 22:00:00' '2022-02-05 23:00:00'
 '2022-02-06 00:00:00' '2022-02-06 01:00:00' '2022-02-06 02:00:00'
 '2022-02-06 03:00:00' '2022-02-06 04:00:00' '2022-02-06 05:00:00'
```

```
 '2022-02-06 06:00:00' '2022-02-06 07:00:00' '2022-02-06 08:00:00'
 '2022-02-06 09:00:00' '2022-02-06 10:00:00' '2022-02-06 11:00:00'
 '2022-02-06 12:00:00' '2022-02-06 13:00:00' '2022-02-06 14:00:00'
 '2022-02-06 15:00:00' '2022-02-06 16:00:00' '2022-02-06 17:00:00'
 '2022-02-06 18:00:00' '2022-02-06 19:00:00' '2022-02-06 20:00:00'
 '2022-02-06 21:00:00' '2022-02-06 22:00:00' '2022-02-06 23:00:00'
 '2022-02-07 00:00:00' '2022-02-07 01:00:00' '2022-02-07 02:00:00'
 '2022-02-07 03:00:00' '2022-02-07 04:00:00' '2022-02-07 05:00:00'
 '2022-02-07 06:00:00' '2022-02-07 07:00:00' '2022-02-07 08:00:00'
 '2022-02-07 09:00:00' '2022-02-07 10:00:00' '2022-02-07 11:00:00'
 '2022-02-07 12:00:00' '2022-02-07 13:00:00' '2022-02-07 14:00:00'
 '2022-02-07 15:00:00' '2022-02-07 16:00:00' '2022-02-07 17:00:00'
 '2022-02-07 18:00:00' '2022-02-07 19:00:00' '2022-02-07 20:00:00'
 '2022-02-07 21:00:00' '2022-02-07 22:00:00' '2022-02-07 23:00:00'
 '2022-02-08 00:00:00' '2022-02-08 01:00:00' '2022-02-08 02:00:00'
 '2022-02-08 03:00:00' '2022-02-08 04:00:00' '2022-02-08 05:00:00'
 '2022-02-08 06:00:00' '2022-02-08 07:00:00' '2022-02-08 08:00:00'
 '2022-02-08 09:00:00' '2022-02-08 10:00:00' '2022-02-08 11:00:00'
 '2022-02-08 12:00:00' '2022-02-08 13:00:00' '2022-02-08 14:00:00'
 '2022-02-08 15:00:00' '2022-02-08 16:00:00' '2022-02-08 17:00:00'
 '2022-02-08 18:00:00' '2022-02-08 19:00:00' '2022-02-08 20:00:00'
 '2022-02-08 21:00:00' '2022-02-08 22:00:00' '2022-02-08 23:00:00'
 '2022-02-09 00:00:00' '2022-02-09 01:00:00' '2022-02-09 02:00:00'
 '2022-02-09 03:00:00' '2022-02-09 04:00:00' '2022-02-09 05:00:00'
 '2022-02-09 06:00:00' '2022-02-09 07:00:00' '2022-02-09 08:00:00'
 '2022-02-09 09:00:00' '2022-02-09 10:00:00' '2022-02-09 11:00:00'
 '2022-02-09 12:00:00' '2022-02-09 13:00:00' '2022-02-09 14:00:00'
 '2022-02-09 15:00:00' '2022-02-09 16:00:00' '2022-02-09 17:00:00'
 '2022-02-09 18:00:00' '2022-02-09 19:00:00' '2022-02-09 20:00:00'
 '2022-02-09 21:00:00' '2022-02-09 22:00:00' '2022-02-09 23:00:00'
 '2022-02-10 00:00:00' '2022-02-10 01:00:00' '2022-02-10 02:00:00'
 '2022-02-10 03:00:00' '2022-02-10 04:00:00' '2022-02-10 05:00:00'
 '2022-02-10 06:00:00' '2022-02-10 07:00:00' '2022-02-10 08:00:00'
 '2022-02-10 09:00:00' '2022-02-10 10:00:00' '2022-02-10 11:00:00'
 '2022-02-10 12:00:00' '2022-02-10 13:00:00' '2022-02-10 14:00:00'
 '2022-02-10 15:00:00' '2022-02-10 16:00:00' '2022-02-10 17:00:00'
 '2022-02-10 18:00:00' '2022-02-10 19:00:00' '2022-02-10 20:00:00'
 '2022-02-10 21:00:00' '2022-02-10 22:00:00' '2022-02-10 23:00:00'
 '2022-02-11 00:00:00' '2022-02-11 01:00:00' '2022-02-11 02:00:00'
 '2022-02-11 03:00:00' '2022-02-11 04:00:00' '2022-02-11 05:00:00'
 '2022-02-11 06:00:00' '2022-02-11 07:00:00' '2022-02-11 08:00:00'
 '2022-02-11 09:00:00' '2022-02-11 10:00:00' '2022-02-11 11:00:00'
 '2022-02-11 12:00:00' '2022-02-11 13:00:00' '2022-02-11 14:00:00'
 '2022-02-11 15:00:00']

HVACUsage : ['On' 'Off']

LightingUsage : ['Off' 'On']

DayOfWeek : ['Monday' 'Saturday' 'Sunday' 'Wednesday' 'Friday'
```

```
'Thursday' 'Tuesday']

Holiday : ['No' 'Yes']
```

2. Objectives

```
df = data.copy()
df
```

```
                Timestamp  Temperature   Humidity  SquareFootage
Occupancy  \
0    2022-01-01 00:00:00    25.139433  43.431581    1565.693999
5
1    2022-01-01 01:00:00    27.731651  54.225919    1411.064918
1
2    2022-01-01 02:00:00    28.704277  58.907658    1755.715009
2
3    2022-01-01 03:00:00    20.080469  50.371637    1452.316318
1
4    2022-01-01 04:00:00    23.097359  51.401421    1094.130359
9
..                   ...          ...        ...            ...
...
995  2022-02-11 11:00:00    28.619382  48.850160    1080.087000
5
996  2022-02-11 12:00:00    23.836647  47.256435    1705.235156
4
997  2022-02-11 13:00:00    23.005340  48.720501    1320.285281
6
998  2022-02-11 14:00:00    25.138365  31.306459    1309.079719
3
999  2022-02-11 15:00:00    23.051165  42.615421    1018.140606
6

    HVACUsage LightingUsage  RenewableEnergy  DayOfWeek Holiday  \
0          On           Off         2.774699     Monday      No
1          On            On        21.831384   Saturday      No
2         Off           Off         6.764672     Sunday      No
3         Off            On         8.623447  Wednesday      No
4          On           Off         3.071969     Friday      No
..        ...           ...              ...        ...     ...
995       Off           Off        21.194696   Saturday      No
996       Off            On        25.748176    Tuesday     Yes
997       Off            On         0.297079     Friday     Yes
998        On           Off        20.425163   Thursday     Yes
999       Off            On         2.455657   Saturday      No

     EnergyConsumption
```

```
0              75.364373
1              83.401855
2              78.270888
3              56.519850
4              70.811732
..                   ...
995            82.306692
996            66.577320
997            72.753471
998            76.950389
999            71.545311

[1000 rows x 11 columns]
```

We will now encode our categorical data.

```
df.replace({'On':1, 'Off':0, 'Yes':1, 'No':0,
            'Monday':1, 'Saturday':6, 'Sunday':7, 'Wednesday':3,
'Friday':5, 'Thursday':4, 'Tuesday':2},inplace=True)
df

               Timestamp  Temperature   Humidity  SquareFootage
Occupancy  \
0    2022-01-01 00:00:00    25.139433  43.431581    1565.693999
5
1    2022-01-01 01:00:00    27.731651  54.225919    1411.064918
1
2    2022-01-01 02:00:00    28.704277  58.907658    1755.715009
2
3    2022-01-01 03:00:00    20.080469  50.371637    1452.316318
1
4    2022-01-01 04:00:00    23.097359  51.401421    1094.130359
9
..                   ...          ...        ...            ...
...
995  2022-02-11 11:00:00    28.619382  48.850160    1080.087000
5
996  2022-02-11 12:00:00    23.836647  47.256435    1705.235156
4
997  2022-02-11 13:00:00    23.005340  48.720501    1320.285281
6
998  2022-02-11 14:00:00    25.138365  31.306459    1309.079719
3
999  2022-02-11 15:00:00    23.051165  42.615421    1018.140606
6

     HVACUsage  LightingUsage  RenewableEnergy  DayOfWeek  Holiday  \
0            1              0         2.774699          1        0
1            1              1        21.831384          6        0
```

```
2            0             0      6.764672          7        0
3            0             1      8.623447          3        0
4            1             0      3.071969          5        0
..         ...           ...           ...        ...      ...
995          0             0     21.194696          6        0
996          0             1     25.748176          2        1
997          0             1      0.297079          5        1
998          1             0     20.425163          4        1
999          0             1      2.455657          6        0

     EnergyConsumption
0            75.364373
1            83.401855
2            78.270888
3            56.519850
4            70.811732
..                 ...
995          82.306692
996          66.577320
997          72.753471
998          76.950389
999          71.545311

[1000 rows x 11 columns]
```

```
df['Timestamp_unix'] =
pd.to_datetime(df['Timestamp']).astype('int64')/10**9
df
```

```
               Timestamp  Temperature   Humidity  SquareFootage
Occupancy  \
0    2022-01-01 00:00:00    25.139433  43.431581    1565.693999
5
1    2022-01-01 01:00:00    27.731651  54.225919    1411.064918
1
2    2022-01-01 02:00:00    28.704277  58.907658    1755.715009
2
3    2022-01-01 03:00:00    20.080469  50.371637    1452.316318
1
4    2022-01-01 04:00:00    23.097359  51.401421    1094.130359
9
..                   ...          ...        ...            ...
...
995  2022-02-11 11:00:00    28.619382  48.850160    1080.087000
5
996  2022-02-11 12:00:00    23.836647  47.256435    1705.235156
4
997  2022-02-11 13:00:00    23.005340  48.720501    1320.285281
6
998  2022-02-11 14:00:00    25.138365  31.306459    1309.079719
```

```
3
999  2022-02-11 15:00:00    23.051165  42.615421    1018.140606
6

     HVACUsage  LightingUsage  RenewableEnergy  DayOfWeek  Holiday  \
0            1              0         2.774699          1        0
1            1              1        21.831384          6        0
2            0              0         6.764672          7        0
3            0              1         8.623447          3        0
4            1              0         3.071969          5        0
..         ...            ...              ...        ...      ...
995          0              0        21.194696          6        0
996          0              1        25.748176          2        1
997          0              1         0.297079          5        1
998          1              0        20.425163          4        1
999          0              1         2.455657          6        0

     EnergyConsumption  Timestamp_unix
0            75.364373    1.640995e+09
1            83.401855    1.640999e+09
2            78.270888    1.641002e+09
3            56.519850    1.641006e+09
4            70.811732    1.641010e+09
..                 ...             ...
995          82.306692    1.644577e+09
996          66.577320    1.644581e+09
997          72.753471    1.644584e+09
998          76.950389    1.644588e+09
999          71.545311    1.644592e+09

[1000 rows x 12 columns]
```

```
df = df.drop(columns='Timestamp')
df
```

```
     Temperature   Humidity  SquareFootage  Occupancy  HVACUsage  \
0      25.139433  43.431581    1565.693999          5          1
1      27.731651  54.225919    1411.064918          1          1
2      28.704277  58.907658    1755.715009          2          0
3      20.080469  50.371637    1452.316318          1          0
4      23.097359  51.401421    1094.130359          9          1
..           ...        ...            ...        ...        ...
995    28.619382  48.850160    1080.087000          5          0
996    23.836647  47.256435    1705.235156          4          0
997    23.005340  48.720501    1320.285281          6          0
998    25.138365  31.306459    1309.079719          3          1
999    23.051165  42.615421    1018.140606          6          0

     LightingUsage  RenewableEnergy  DayOfWeek  Holiday
EnergyConsumption  \
```

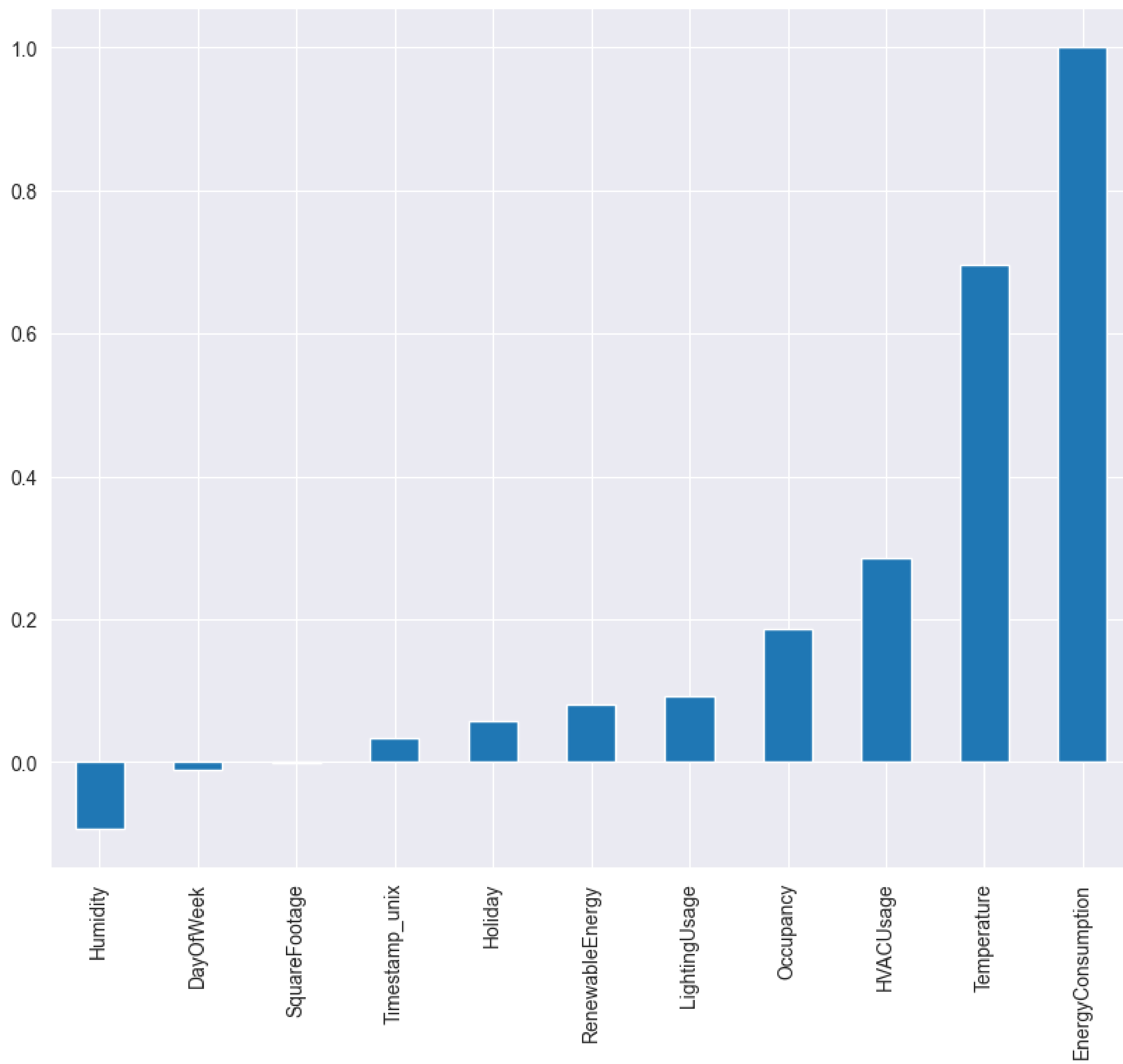
```
0                 0         2.774699          1         0
75.364373
1                 1        21.831384          6         0
83.401855
2                 0         6.764672          7         0
78.270888
3                 1         8.623447          3         0
56.519850
4                 0         3.071969          5         0
70.811732
..              ...              ...        ...       ...
...
995               0        21.194696          6         0
82.306692
996               1        25.748176          2         1
66.577320
997               1         0.297079          5         1
72.753471
998               0        20.425163          4         1
76.950389
999               1         2.455657          6         0
71.545311

     Timestamp_unix
0      1.640995e+09
1      1.640999e+09
2      1.641002e+09
3      1.641006e+09
4      1.641010e+09
..              ...
995    1.644577e+09
996    1.644581e+09
997    1.644584e+09
998    1.644588e+09
999    1.644592e+09

[1000 rows x 11 columns]
```

```
df.EnergyConsumption.skew()
```

```
0.027398907453860765
```

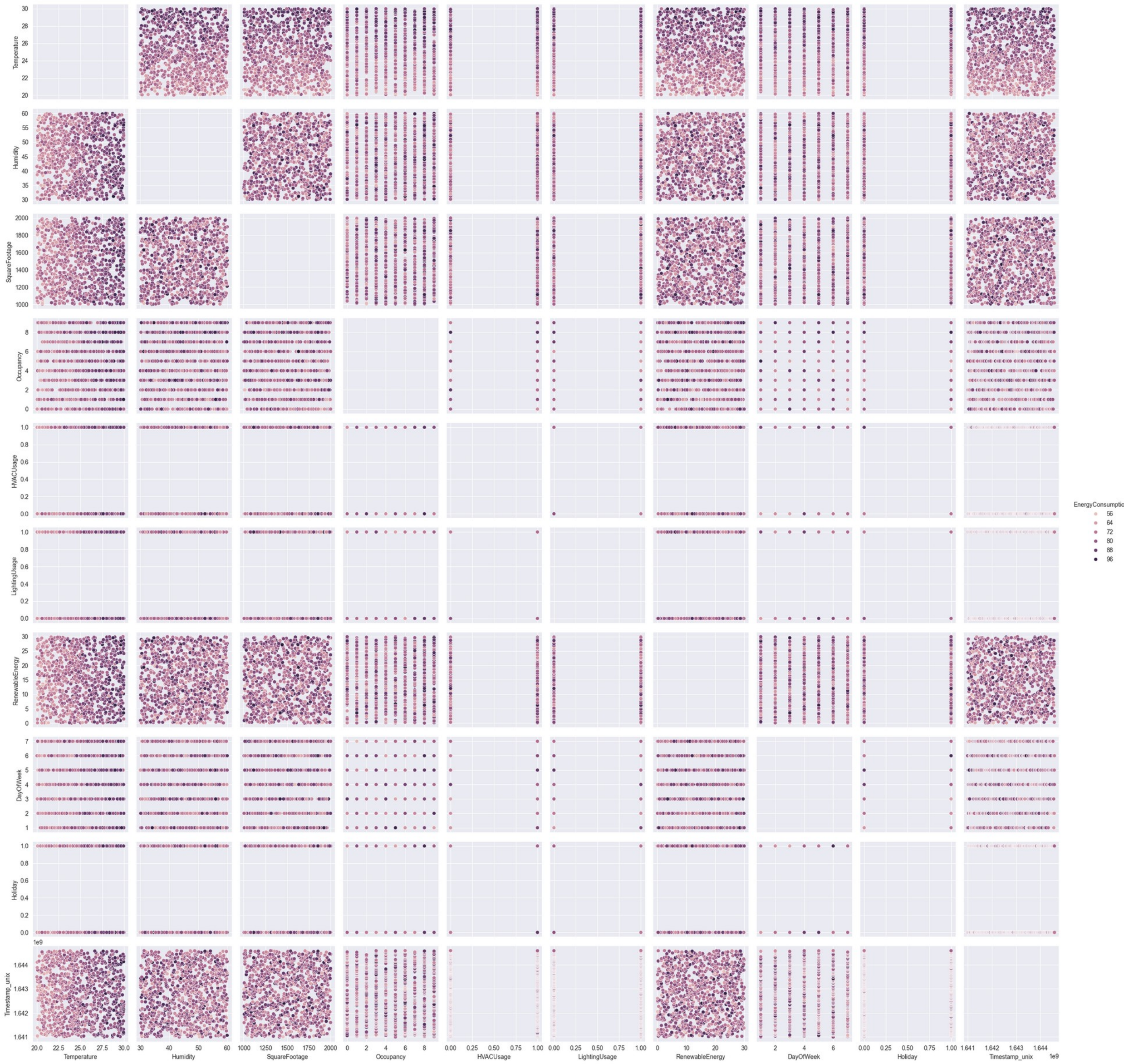
```
corelations = df.corr()['EnergyConsumption'].sort_values()
corelations.plot(kind='bar', figsize=(10,8))
```

```
<Axes: >
```




```python
sns.pairplot(df, hue='EnergyConsumption')
```

```
<seaborn.axisgrid.PairGrid at 0x1dcf0a35c10>
```



Data visualisation and encoding is important for representation of data and preparation for our prediction models.

After cleaning and encoding our data we can start implementing couple regression models. After we do that we will check their efficiency in predicting values on our holdout set (we will set it to 10% of the whole dataframe).

3. Linear Regression Models

```
X = df.drop('EnergyConsumption',axis=1)
y = df['EnergyConsumption']

skb = SelectKBest(k=4,score_func=f_regression)
transX = skb.fit_transform(X,y)
```

```
srt_skb = skb.pvalues_.argsort()[::-1]
print(srt_skb)
skb.pvalues_
```

```
[2 7 9 8 6 5 1 3 4 0]
```

```
array([5.70222544e-146, 3.05498261e-003, 9.71553934e-001, 2.76301328e-009,
       2.42306979e-020, 3.10566753e-003, 1.02428573e-002, 7.39582072e-001,
       7.32207624e-002, 2.77040556e-001])
```

```
skb.feature_names_in_
```

```
array(['Temperature', 'Humidity', 'SquareFootage', 'Occupancy',
       'HVACUsage', 'LightingUsage', 'RenewableEnergy', 'DayOfWeek',
       'Holiday', 'Timestamp_unix'], dtype=object)
```

```
X_SKB = df[['Temperature', 'HVACUsage', 'Occupancy',
'Humidity','LightingUsage']]
X_SKB
```

	Temperature	HVACUsage	Occupancy	Humidity	LightingUsage
0	25.139433	1	5	43.431581	0
1	27.731651	1	1	54.225919	1
2	28.704277	0	2	58.907658	0
3	20.080469	0	1	50.371637	1
4	23.097359	1	9	51.401421	0
..	...	...	...	...	...
995	28.619382	0	5	48.850160	0
996	23.836647	0	4	47.256435	1
997	23.005340	0	6	48.720501	1
998	25.138365	1	3	31.306459	0
999	23.051165	0	6	42.615421	1

```
[1000 rows x 5 columns]
```

SelectKBest p-values shows su that all of our features are significant, top five are selected in `X_SKB` above.

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.1,random_state=72018)


Input1=[('polynomial', PolynomialFeatures(include_bias=False)),
('ss',StandardScaler() ),('model',Lasso(tol = 0.2))]
pipe1 = Pipeline(Input1)

param_grid1 = {
    "polynomial__degree": [ 1, 2,3,4,5],
    "model__alpha":[0.0001,0.001,0.01,0.1,1,10,100,1000]
```

```
}
search1 = GridSearchCV(pipe1, param_grid1, n_jobs=2)
search1.fit(X_train, y_train)
best1=search1.best_estimator_
print(search1.best_params_)
best1
```

```
{'model__alpha': 0.1, 'polynomial__degree': 1}
```

```
Pipeline(steps=[('polynomial',
                  PolynomialFeatures(degree=1, include_bias=False)),
                 ('ss', StandardScaler()),
                 ('model', Lasso(alpha=0.1, tol=0.2))])
```

```
Input2=[('polynomial', PolynomialFeatures(include_bias=False)),
('ss',StandardScaler() ), ('model',Ridge())]
pipe2 = Pipeline(Input2)

param_grid2 = {
    "polynomial__degree": [ 1, 2,3,4,5],
    "model__alpha":[0.0001,0.001,0.01,0.1,1,10,100,1000]
}

search2 = GridSearchCV(pipe2, param_grid2, n_jobs=2)
search2.fit(X_train, y_train)
best2=search2.best_estimator_
print(search2.best_params_)
best2
```

```
{'model__alpha': 10, 'polynomial__degree': 1}
```

```
Pipeline(steps=[('polynomial',
                  PolynomialFeatures(degree=1, include_bias=False)),
                 ('ss', StandardScaler()), ('model', Ridge(alpha=10))])
```

```
Input3 = [('polynomial', PolynomialFeatures(include_bias=False)),
('ss', StandardScaler()), ('model',ElasticNet(tol=0.2))]
pipe3 = Pipeline(Input3)

param_grid3 = {
    "polynomial__degree": [ 1, 2,3,4,5],
    "model__alpha":[0.0001,0.001,0.01,0.1,1,10,100,1000],
    "model__l1_ratio":[0.0001,0.001,0.01,0.1,1,10,100,1000]
}

search3 = GridSearchCV(pipe3, param_grid3, n_jobs=2)
search3.fit(X_train, y_train)
best3=search3.best_estimator_
```

```
print(search3.best_params_)
best3
```

```
{'model__alpha': 0.1, 'model__l1_ratio': 1, 'polynomial__degree': 1}
```

```
Pipeline(steps=[('polynomial',
                  PolynomialFeatures(degree=1, include_bias=False)),
                 ('ss', StandardScaler()),
                 ('model', ElasticNet(alpha=0.1, l1_ratio=1,
tol=0.2))])
```

```
print('for the training set R^2 of the best estimators for Lasso,
Ridge and ElasticNet (in that order) are:')
print(best1.score(X_train,y_train))
print(best2.score(X_train,y_train))
print(best3.score(X_train,y_train))
```

```
for the training set R^2 of the best estimators for Lasso, Ridge and
ElasticNet (in that order) are:
0.6111921987614747
0.6124810926429878
0.6111921987614747
```

```
print('for the test set R^2 of the best estimators for Lasso, Ridge
and ElasticNet (in that order) are:')
print(best1.score(X_test,y_test))
print(best2.score(X_test,y_test))
print(best3.score(X_test,y_test))
```

```
for the test set R^2 of the best estimators for Lasso, Ridge and
ElasticNet (in that order) are:
0.6645368081717871
0.6686871143154163
0.6645368081717871
```

```
Input4 = [('ss', StandardScaler()), ('model',LinearRegression())]
pipe4 = Pipeline(Input4)
```

```
pipe4.fit(X_train, y_train)
```

```
Pipeline(steps=[('ss', StandardScaler()), ('model',
LinearRegression())])
```

```
pipe4.score(X_train, y_train)
```

```
0.6125550867319011
```

```
pipe4.score(X_test, y_test)
```

```
0.6700805211563543
```

```
print(mean_squared_error(best1.predict(X_train), y_train))
```

```
25.245331859788127

print(mean_squared_error(best2.predict(X_train), y_train))

25.161643843068298

print(mean_squared_error(best3.predict(X_train), y_train))

25.245331859788127

print(mean_squared_error(pipe4.predict(X_train), y_train))

25.15683939901052
```

All above outputs present R^2 score as well as MSE score for Lasso, Ridge, ElasticNet and Linear regression models. Maximum prediction score from all of the models is 0.67 for the test set of data and 0.61 when predicting on train set.

Those scores are low but there should be a model that predicts more accurately for this specific dataset. That model is not amongst these above because even after preforming grid search we could not find a good enough model.

In above outputs we can also see the best estimators with their respective best hyperparameters picked from `param_grid` for each model (except `LinearRegression()` model).

4. Insights and key findings

The chosen dataset proves that linear regression is sometimes not enough to make the most accurate predictions. Although that is true, we can still see that our linear models have scored above 60% in predicting the values of the holdout set.

We also need to keep in mind that all features of the set are to some degree statistically significant to the target value outcomes. From above histogram we se strong correlations between features.

Finally, this dataset is consisted of enough information and there is no missing or duplicated values that would hinder our possibilities to predict the target values.

Our `MSE` and `R^2` scores are not that high but not so low, they are somewhere in the middle meaning that we didn't find the best model for this dataset, but we did not miss completely with our models as well.

5. Next Steps

Next step would be to find some other model or models that would suit better to this dataset. We would need to explore more models and tune them with their respective hyperparameters to find the best estimator using grid search.

Generally speaking this project shows only what can be called a great introduction to some linear models.