

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import RidgeCV
import plotly.express as px
```

```
In [2]: # loading the dataset
df = pd.read_csv("Users/kumarohit/Downloads/Lasso and Ridge Regression dataset.csv")
df
```

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2
0	0.98	514.5	294.0	110.25	7.0	2.0	0.0	0.0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3.0	0.0	0.0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4.0	0.0	0.0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5.0	0.0	0.0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2.0	0.0	0.0	20.84	28.28
...	...	...	...	...	...	...	...	...	...	...
1291	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1292	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1293	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1294	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1295	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1296 rows x 10 columns

```
In [3]: df.isnull().sum()
```

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2
0	0.98	514.5	294.0	110.25	7.0	2.0	0.0	0.0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3.0	0.0	0.0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4.0	0.0	0.0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5.0	0.0	0.0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2.0	0.0	0.0	20.84	28.28
...	...	...	...	...	...	...	...	...	...	...
1291	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1292	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1293	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1294	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1295	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1296 rows x 10 columns

```
In [4]: df
```

```
Out[4]:
```

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2
0	0.98	514.5	294.0	110.25	7.0	2.0	0.0	0.0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3.0	0.0	0.0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4.0	0.0	0.0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5.0	0.0	0.0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2.0	0.0	0.0	20.84	28.28
...	...	...	...	...	...	...	...	...	...	...
1291	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1292	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1293	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1294	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1295	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1296 rows x 10 columns

```
In [5]: # Changing the names of the columns
df.columns = ['Relative Compactness',
'Surface Area',
'Wall Area',
'Roof Area',
'Overall Height',
'Orientation',
'Glazing Area',
'Glazing Area Distribution',
'Heating Load',
'Cooling Load']
```

```
In [6]: df
```

```
Out[6]:
```

	Relative Compactness	Surface Area	Wall Area	Roof Area	Overall Height	Orientation	Glazing Area	Glazing Area Distribution	Heating Load	Cooling Load
0	0.98	514.5	294.0	110.25	7.0	2.0	0.0	0.0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3.0	0.0	0.0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4.0	0.0	0.0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5.0	0.0	0.0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2.0	0.0	0.0	20.84	28.28
...	...	...	...	...	...	...	...	...	...	...
1291	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1292	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1293	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1294	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1295	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

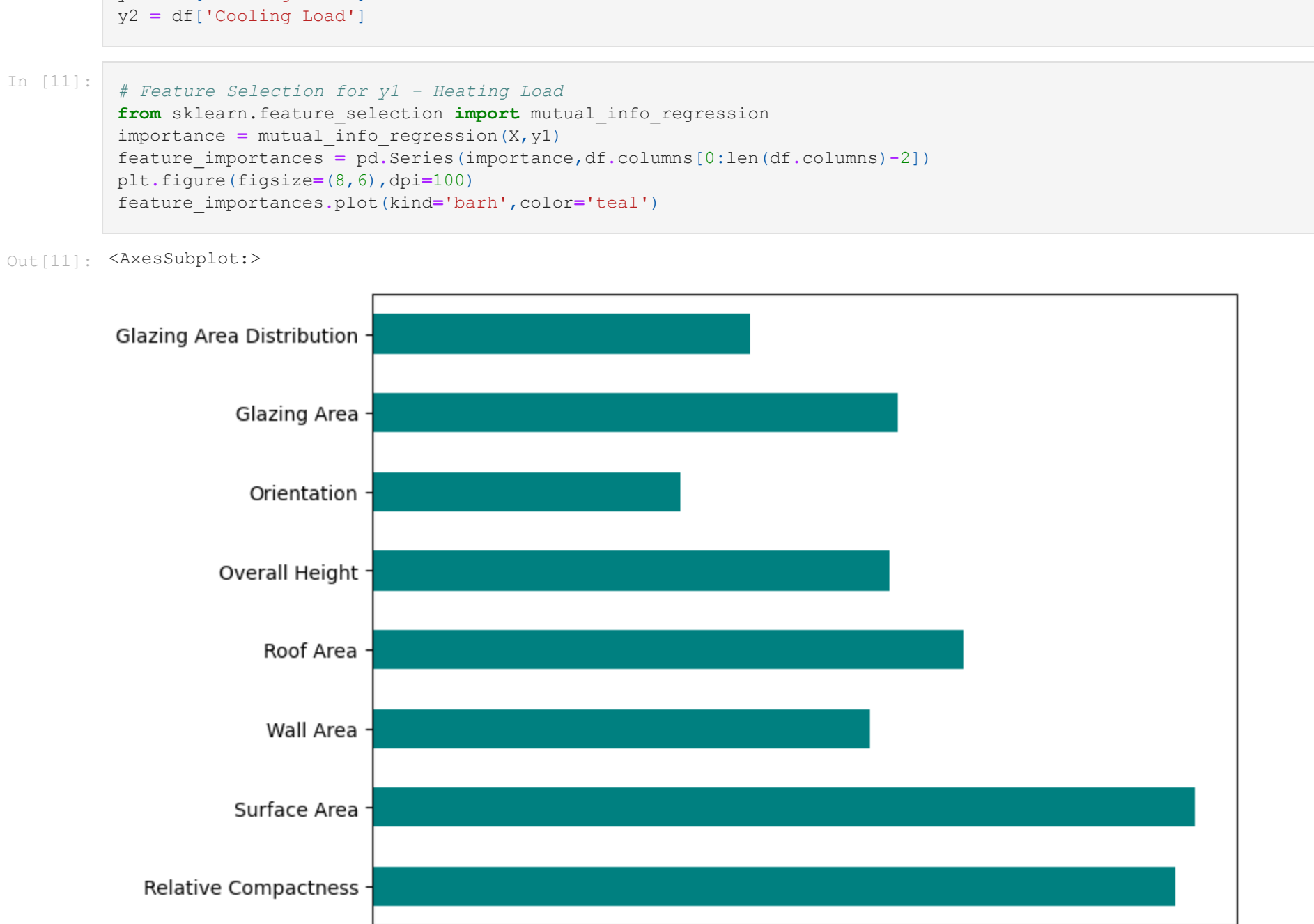
1296 rows x 10 columns

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1296 entries, 0 to 1295
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   Relative Compactness  768 non-null    float64
1   Surface Area          768 non-null    float64
2   Wall Area             768 non-null    float64
3   Roof Area             768 non-null    float64
4   Overall Height        768 non-null    float64
5   Orientation            768 non-null    float64
6   Glazing Area          768 non-null    float64
7   Glazing Area Distribution 768 non-null    float64
8   Heating Load          768 non-null    float64
9   Cooling Load          768 non-null    float64
dtypes: float64(10)
memory usage: 101.4 KB
```

```
In [8]: df = df.fillna(df.mean())
```

```
In [71]: plt.figure(figsize=(10,10),dpi=100)
sns.heatmap(df.corr(),annot=True,cmap='YlGnBu')
```



```
In [10]: X = df.drop(['Heating Load','Cooling Load'],axis=1)
y1 = df['Heating Load']
y2 = df['Cooling Load']
```

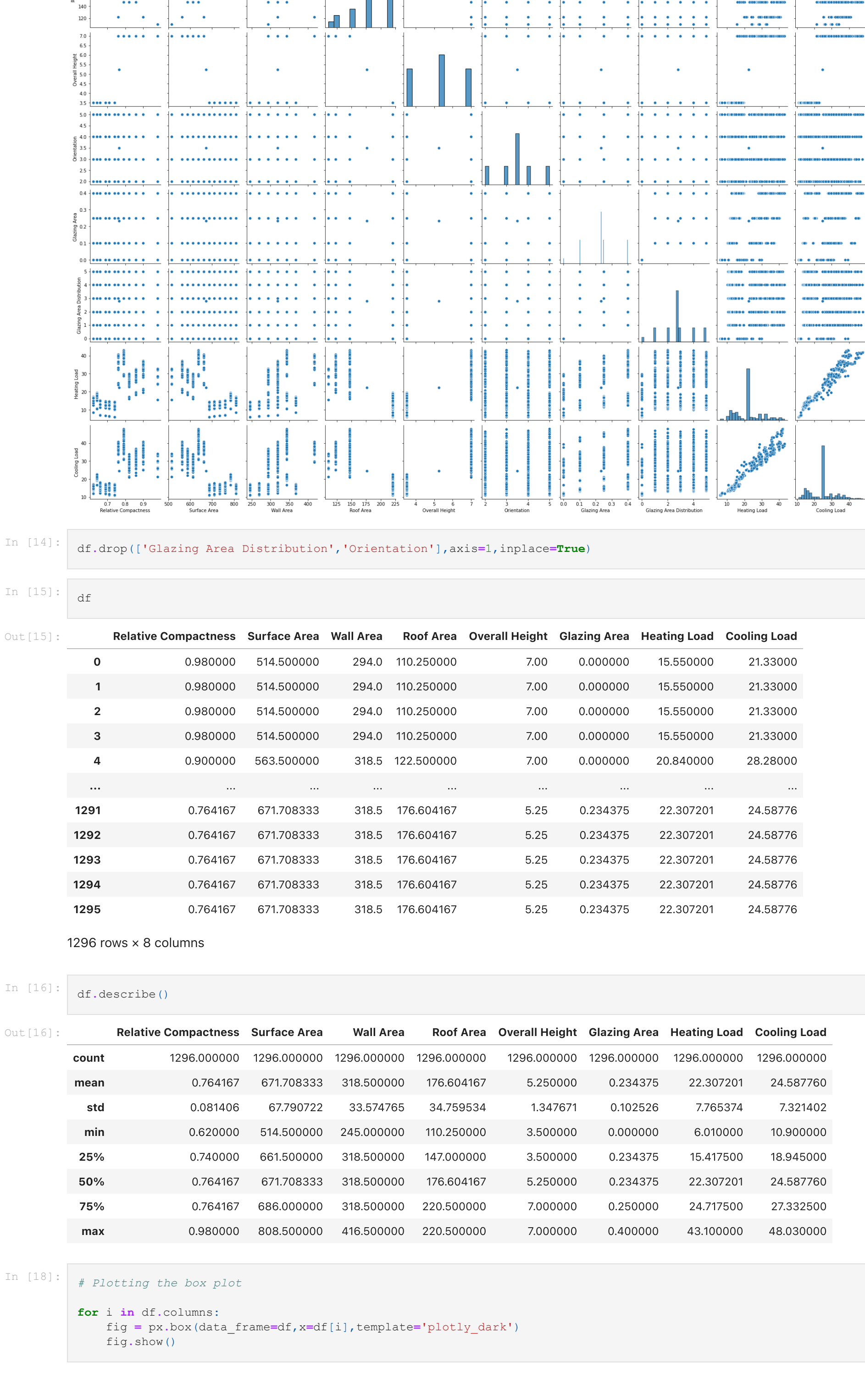
```
In [11]: # Feature Selection for y1 = Heating Load
from sklearn.feature_selection import mutual_info_regression
importance = mutual_info_regression(X,y1)
feature_importances = pd.Series(importance,df.columns[0:len(df.columns)-2])
plt.figure(figsize=(8,6),dpi=100)
feature_importances.plot(kind='barh',color='teal')
```

Out[11]: <AxesSubplot>

```
In [12]: # Feature Selection for y2 = Cooling Load
from sklearn.feature_selection import mutual_info_regression
importance = mutual_info_regression(X,y2)
feature_importances = pd.Series(importance,df.columns[0:len(df.columns)-2])
plt.figure(figsize=(8,6),dpi=100)
feature_importances.plot(kind='barh',color='teal')
```

Out[12]: <AxesSubplot>

```
In [13]: plt.figure(figsize=(10,10),dpi=100)
sns.pairplot(data=df)
```



```
In [14]: df.drop(['Glazing Area Distribution','Orientation'],axis=1,inplace=True)
```

```
In [15]: df
```

	Relative Compactness	Surface Area	Wall Area	Roof Area	Overall Height	Glazing Area	Heating Load	Cooling Load
0	0.980000	514.500000	294.0	110.250000	7.00	0.000000	15.550000	21.33000
1	0.980000	514.500000	294.0	110.250000	7.00	0.000000	15.550000	21.33000
2	0.980000	514.500000	294.0	110.250000	7.00	0.000000	15.550000	21.33000
3	0.980000	514.500000	294.0	110.250000	7.00	0.000000	15.550000	21.33000
4	0.900000	563.500000	318.5	122.500000	7.00	0.000000	20.840000	28.28000
...	...	...	...	...	...	...	...	...
1291	0.764167	671.708333	318.5	176.604167	5.25	0.234375	22.307201	24.58776
1292	0.764167	671.708333	318.5	176.604167	5.25	0.234375	22.307201	24.58776
1293	0.764167	671.708333	318.5	176.604167	5.25	0.234375	22.307201	24.58776
1294	0.764167	671.708333	318.5	176.604167	5.25	0.234375	22.307201	24.58776
1295	0.764167	671.708333	318.5	176.604167	5.25	0.234375	22.307201	24.58776

1296 rows x 8 columns

```
In [16]: df.describe()
```

	Relative Compactness	Surface Area	Wall Area	Roof Area	Overall Height	Glazing Area	Heating Load	Cooling Load
count	1296.000000	1296.000000	1296.000000	1296.000000	1296.000000	1296.000000	1296.000000	1296.000000
mean	0.764167	671.708333	318.500000	176.604167	5.250000	0.234375	22.307201	24.587760
std	0.081406	67.790722	33.574765	34.769534	1.347671	0.102526	7.765374	7.321402
min	0.620000	514.500000	245.000000	110.250000	3.500000	0.000000	6.010000	10.900000
25%	0.740000	661.500000	318.500000	147.000000	3.500000	0.234375	16.417500	18.945000
50%	0.764167	671.708333	318.500000	176.604167	5.250000	0.234375	22.307201	24.587760
75%	0.764167	686.000000	318.500000	220.500000	7.000000	0.250000	24.717500	27.332500
max	0.980000	808.500000	416.500000	270.500000	7.000000	0.400000	43.100000	48.030000

```
In [18]: # Plotting the box plot
for i in df.columns:
fig = px.box(data_frame=df,x=df[i],template='plotly_dark')
fig.show()
```

```
In [21]: # Checking for the unique values
for i in df.columns:
print(f"The unique values for the column {i} are", df[i].unique(), "\n")
```



