Data Science Project

Python Report

Prepared by Group 6

Team members

Bhavana Nare

Vishnu Priya Buggineni

Yaswanth Kumar Palli

Vamshi Krishna B

Regression Models:

Linear Regression
Ridge Regression
Lasso Ridge
Quadratic Regression
Symbolic Regression
Symbolic Ridge Regression
Symbolic Lasso Ridge

Main Points

For Linear Regression Model Used SFS function for Forward, Backward and Stepwise
For Ridge we have used RidgeCV
For Lasso We have used LassoCV
For Quadratic PolynomialRegression with degree 2
For Symbolic Based on datasets Power values are considered

We have worked on all 6 datasets using both **python** and **scala**.
All the libraries we used in python are mentioned below:
Numpy, Pandas, Matplotlib, Seaborn, Mlxtend, Gplearn, Math, and sklearn

Air Quality Dataset:

We have taken this dataset from UCI repository(we used AirQualityUCI_1.csv in the datasets for python) . There are 9358 occurrences of hourly averaged responses from an array of 5 metal oxide chemical sensors integrated in an Air Quality Chemical Multisensor Device in this dataset. This dataset contains 12 attributes, the information about each attribute is mentioned below:

Date- format (DD/MM/YYYY)
Time- format (HH.MM.SS)
CO(GT)- True hourly averaged concentration CO
PT08.S1(CO)- (tin oxide) hourly averaged sensor response
NMHC(GT)- True hourly averaged overall Non Metanic HydroCarbons concentration
C6H6(GT)- True hourly averaged Benzene concentration
PT08.S2(NMHC)- (titania) hourly averaged sensor response
NOx(GT)- True hourly averaged NOx concentration
PT08.S3(NOx)- (tungsten oxide) hourly averaged sensor response

NO2(GT)- True hourly averaged NO2 concentration—-**Response Variable**
PT08.S4(NO2)- (tungsten oxide) hourly averaged sensor response
PT08.S5(O3) -(indium oxide) hourly averaged sensor response
T- Temperature in °C
RH- Relative Humidity (%)
AH- Absolute Humidity

Python report:
The steps followed are described below:

1. We have imported all the necessary libraries
2. Loading the dataset
3. Renamed all the attributes as mentioned below:
   "CO(GT)":"CO_Concentrate",
   "PT08.S1(CO)": "Tin_Oxide",
   "NMHC(GT)": "Non_Metanic_Hydrocarbons",
   "C6H6(GT)": "Benzene_Concentration",
   "PT08.S2(NMHC)":"Titania_Concentration",
   "NOx(GT)":"NOx",
   "PT08.S3(NOx)":"Tungsten_Oxide_NOx",
   "NO2(GT)":"NO2",
   "PT08.S4(NO2)":"Tungsten_Oxide_NO2",
   "PT08.S5(O3)":"Indium_Oxide",
   "T":"Temperature",
   "RH":"Relative_Humidity",
   "AH":"Absolute_Humidity"
4. Checking all the null values

```
In [6]: data_1.isnull().sum()

Out[6]: Date                        0
        Time                        0
        CO_Concentrate              0
        Tin_Oxide                   0
        Non_Metanic_Hydrocarbons    0
        Benzene_Concentration       0
        Titania_Concentration       0
        NOx                         0
        Tungsten_Oxide_NOx          0
        NO2                         0
        Tungsten_Oxide_NO2          0
        Indium_Oxide                0
        Temperature                 0
        Relative_Humidity           0
        Absolute_Humidity           0
        dtype: int64
```

There are no null values in the airquality dataset

5. Checking the presence of negative values in the data

```
data_1.describe()
```

	CO_Concentrate	Tin_Oxide	Non_Metanic_Hydrocarbons	Benzene_Concentration	Titania_Concentration	NOx	Tungsten_Oxide_NOx	NO
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.00000
mean	-34.207524	1048.869652	-159.090093	1.865576	894.475963	168.604200	794.872333	58.13589
std	77.657170	329.817015	139.789093	41.380154	342.315902	257.424561	321.977031	126.93142
min	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.00000
25%	0.600000	921.000000	-200.000000	4.004958	711.000000	50.000000	637.000000	53.00000
50%	1.500000	1052.500000	-200.000000	7.886653	894.500000	141.000000	794.250000	96.00000
75%	2.600000	1221.250000	-200.000000	13.636091	1104.750000	284.200000	960.250000	133.00000
max	11.900000	2039.750000	1189.000000	63.741476	2214.000000	1479.000000	2682.750000	339.70000

We can clearly see that the minimum value of each feature is -200. We can replace negative values in the data with zero, mean, median etc

6. Replacing the negative values with zero. The description of all attributes after replacing negative values is mentioned below:

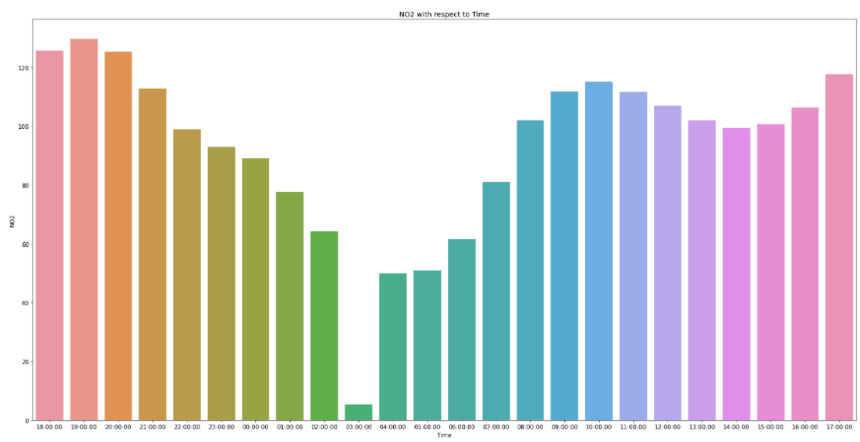
```
In [9]: data_1.describe()
Out[9]:
```

	CO_Concentrate	Tin_Oxide	Non_Metanic_Hydrocarbons	Benzene_Concentration	Titania_Concentration	NOx	Tungsten_Oxide_NOx	NO
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.00000
mean	1.765545	1056.692672	21.373731	9.688596	902.298983	203.636796	802.695353	93.23261
std	1.554264	301.232260	91.103489	7.559609	318.681183	214.984126	299.341439	61.46858
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	0.600000	921.000000	0.000000	4.004958	711.000000	50.000000	637.000000	53.00000
50%	1.500000	1052.500000	0.000000	7.886653	894.500000	141.000000	794.250000	96.00000
75%	2.600000	1221.250000	0.000000	13.636091	1104.750000	284.200000	960.250000	133.00000
max	11.900000	2039.750000	1189.000000	63.741476	2214.000000	1479.000000	2682.750000	339.70000

We can also replace the negative values with mean, median etc

7. Plotting Time Vs NO2. A bar plot for NO2 with respect to time is shown below:

```
Out[33]: Text(0.5, 1.0, 'NO2 with respect to Time')
```



8. Forward Feature Selection, The best feature selected using forward feature selection is mentioned below:

```
Selected Features : ('CO_Concentrate', 'Tin_Oxide', 'Benzene_Concentration', 'NOx', 'Indium_Oxide', 'Relative_Humidity')
Selected Features ID : (0, 1, 3, 5, 8, 10)

[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.0s finished
```

9. Backward Feature Selection, The best feature selected using backward feature selection is mentioned below:

```
Selected Features : ('CO_Concentrate', 'Benzene_Concentration', 'Titania_Concentration', 'NOx', 'Indium_Oxide', 'Relative_Humid
ity')
Selected Features ID : (0, 3, 4, 5, 8, 10)

[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
```

10. Step-wise Feature Selection, The best feature selected using step-wise feature selection is mentioned below:

```
Selected Features : ('CO_Concentrate', 'Tin_Oxide', 'Benzene_Concentration', 'NOx', 'Indium_Oxide', 'Relative_Humidity')
Selected Features ID : (0, 1, 3, 5, 8, 10)

[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s finished
```

11. We selected the features selected from Forward feature selection and created a new Dataframe using the features selected. We used the newly created dataframe in all the regression model.
12. Splitting the new dataframe into train and test sets using the train_test_split
13. Running all the regression models(linear regression, Ridge regression, Lasso Regression, Polynomial regression, symbolic regression)

Summary of all the models:

a. **Linear Regression**
   R2: 0.7409648928470239
   Adj R2: 1.064741475485335

b. **Ridge Regression:**
   R2: 0.7409660836331564
   Adj R2: 1.0647411778683362

c. **Lasso Regression:**
   R2:0.7195064365162228
   Adj R2: 1.0701046563289003

d. **Quadratic Regression:**
   R2: 0.8389213177239685
   Adj R2: 1.0402589119073544

e. **Symbolic Regression:**
   R2: 0.7807531306825347
   Adj R2: 1.0547970735363315

   Quadratic Regression works best when we replace negative values with zero based on R2.

ForestFires Dataset:
We have taken this dataset from the UCI repository. Our aim is to predict the burned area of the forest. There are 518 instances in the dataset.

1. X - x-axis spatial coordinate within the Montesinho park map: 1 to 9
2. Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9
3. month - month of the year: 'jan' to 'dec'
4. day - day of the week: 'mon' to 'sun'
5. FFMC - FFMC index from the FWI system: 18.7 to 96.20
6. DMC - DMC index from the FWI system: 1.1 to 291.3
7. DC - DC index from the FWI system: 7.9 to 860.6
8. ISI - ISI index from the FWI system: 0.0 to 56.10
9. temp - temperature in Celsius degrees: 2.2 to 33.30
10. RH - relative humidity in %: 15.0 to 100
11. wind - wind speed in km/h: 0.40 to 9.40
12. rain - outside rain in mm/m2 : 0.0 to 6.4
13. area - the burned area of the forest (in ha): 0.00 to 1090.84

(this output variable is very skewed towards 0.0, thus it may make sense to model with the logarithm transform).

Libraries used in Python:

Numpy
Pandas
Matplotlib
Seaborn
Sklearn

------------------ Displaying head ------------------

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0

------------------ Contents of dataset ------------------

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 13 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   X       517 non-null    int64
 1   Y       517 non-null    int64
 2   month   517 non-null    object
 3   day     517 non-null    object
 4   FFMC    517 non-null    float64
 5   DMC     517 non-null    float64
 6   DC      517 non-null    float64
 7   ISI     517 non-null    float64
 8   temp    517 non-null    float64
 9   RH      517 non-null    int64
 10  wind    517 non-null    float64
 11  rain    517 non-null    float64
 12  area    517 non-null    float64

dtypes: float64(8), int64(3),  object(2)
```

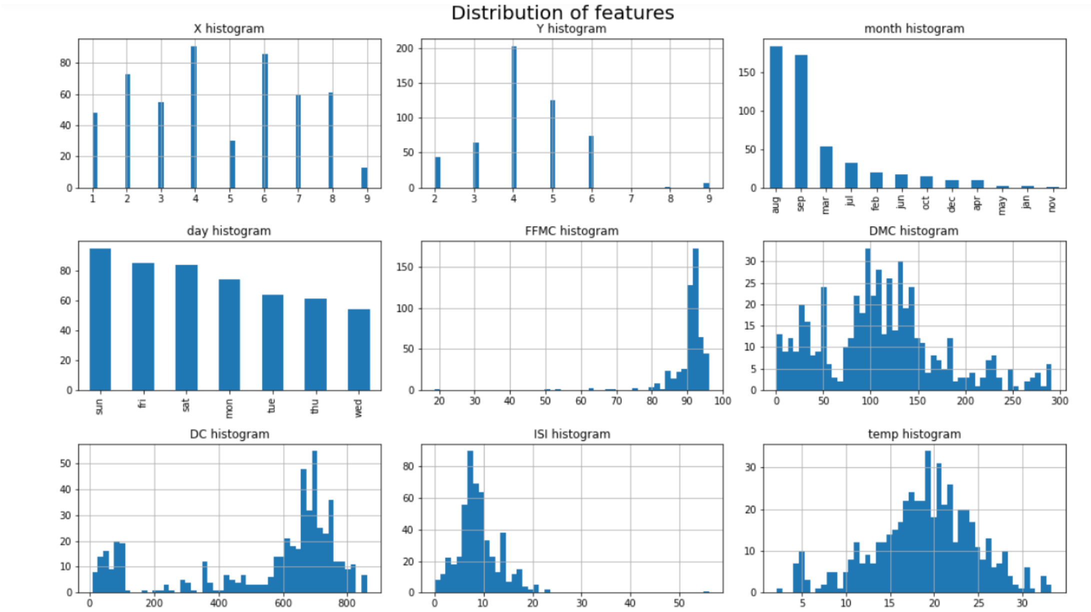
Variables month and day being ordinal variables, they have been encoded using label encoding.

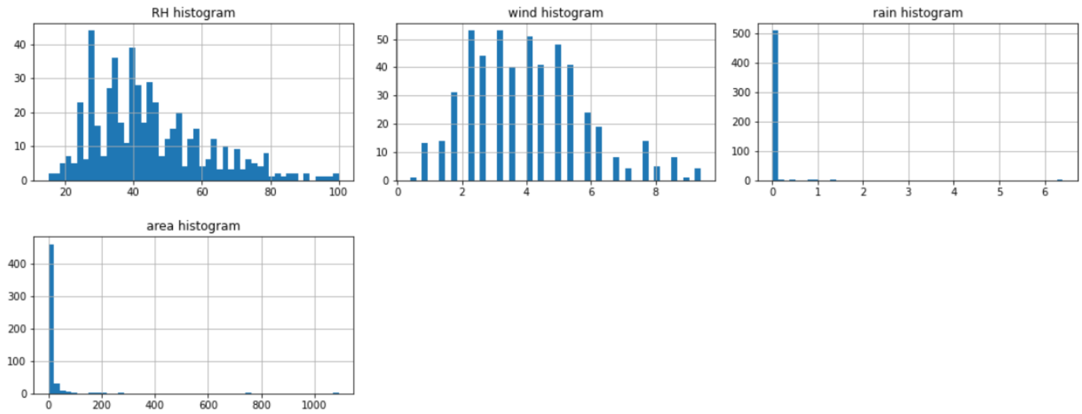
No Null values were detected in the data.
Response variables are highly positively skewed and most of the values in this variable are zeros. Which makes this regression exercise very difficult and even the R-squared and Adj R-squared values are very low.

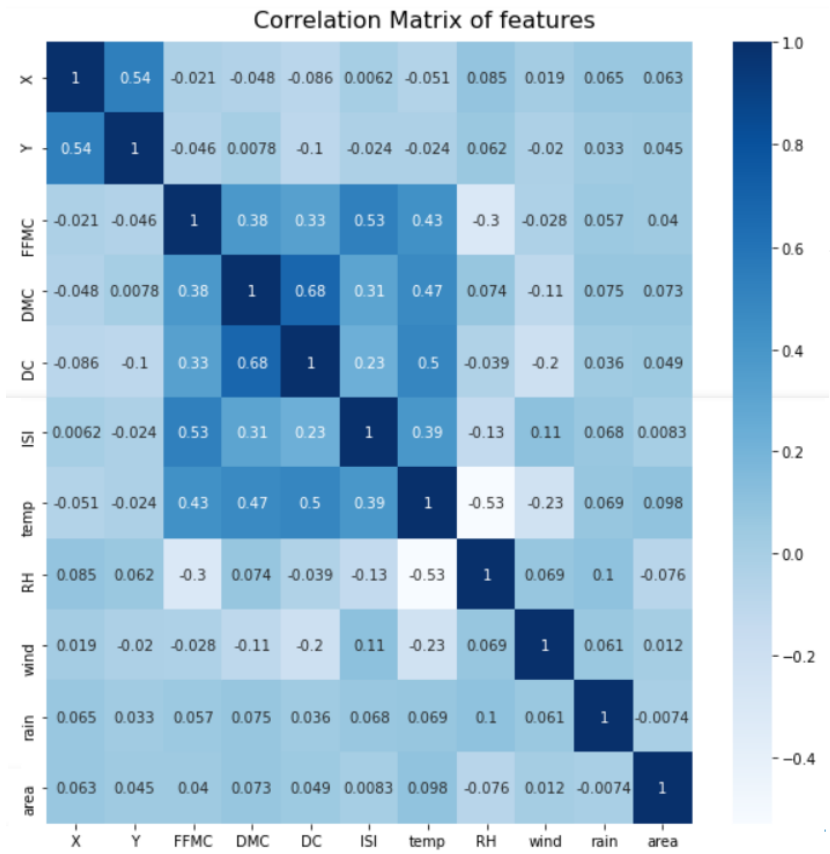
Custom summary of numerical features.

	Feature_name	datatype	Count	min	quartile1	Mean	Median	quartile3	max	Std dev	Skewness	Kurtosis	Range	IQR	skewness comment	outlier comment
0	X	int64	517	1.0	3.0	4.669246	4.00	7.00	9.00	2.31	0.04	-1.17	8.00	4.00	Fairly symmetric(positive)	No outliers
1	Y	int64	517	2.0	4.0	4.299807	4.00	5.00	9.00	1.23	0.42	1.42	7.00	1.00	Fairly symmetric(positive)	Has outliers
2	FFMC	float64	517	18.7	90.2	90.644681	91.60	92.90	96.20	5.52	-6.58	67.07	77.50	2.70	High negative skewed	Has outliers
3	DMC	float64	517	1.1	68.6	110.872340	108.30	142.40	291.30	64.05	0.55	0.20	290.20	73.80	Moderate positive skewed	Has outliers
4	DC	float64	517	7.9	437.7	547.940039	664.20	713.90	860.60	248.07	-1.10	-0.25	852.70	276.20	High negative skewed	Has outliers
5	ISI	float64	517	0.0	6.5	9.021663	8.40	10.80	56.10	4.56	2.54	21.46	56.10	4.30	High positive skewed	Has outliers
6	temp	float64	517	2.2	15.5	18.889168	19.30	22.80	33.30	5.81	-0.33	0.14	31.10	7.30	Fairly symmetric(negative)	Has outliers
7	RH	int64	517	15.0	33.0	44.288201	42.00	53.00	100.00	16.32	0.86	0.44	85.00	20.00	Moderate positive skewed	Has outliers
8	wind	float64	517	0.4	2.7	4.017602	4.00	4.90	9.40	1.79	0.57	0.05	9.00	2.20	Moderate positive skewed	Has outliers
9	rain	float64	517	0.0	0.0	0.021663	0.00	0.00	6.40	0.30	19.82	421.30	6.40	0.00	High positive skewed	Has outliers
10	area	float64	517	0.0	0.0	12.847292	0.52	6.57	1090.84	63.66	12.85	194.14	1090.84	6.57	High positive skewed	Has outliers

Distribution of all features







Feature selection
From the three feature selection techniques forward, backward and stepwise, all the 12 independent variables were selected.

After the feature selection, scaling was performed on the independent datasets using standard scalar.

Model results.
************** R^2 and R-bar squared of Models **************
Linear Regression: R^2 score is 0.021557347582511044
Linear Regression: Adjusted R^2 score is 0.00024473535163505034

Best Alpha 4.863009016651023

Lasso Regression: R^2 score is -6.493504504101466e-06
Lasso Regression: Adjusted R^2 score is -0.021788813164998366

Best Alpha 0.9899999999999995
Ridge Regression: R^2 score is 0.021558995633138567
Ridge Regression: Adjusted R^2 score is 0.0002464193003950399

Quadratic Regression: R^2 score is -9.640437028798407e+29
Quadratic Regression: Adjusted R^2 score is -9.850426746257383e+29
************** R^2-cross-validated of Models **************

Linear Regression R^2 score is: 0.001 (0.009)

Lasso Regression R^2 score is: 0.006 (0.002)

Ridge Regression R^2 score is: 0.003 (0.006)

Quadratic Regression R^2 score is: 0.001 (0.009)

Symbolic Regression R^2 score is: -0.042 (0.016)

Expedia Dataset:

```
------------------ Contents of dataset  ------------------
RangeIndex: 2870 entries, 0 to 2869
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   2870 non-null   int64
 1   region               2870 non-null   object
 2   latitude             2870 non-null   float64
 3   longitude            2870 non-null   float64
 4   accommodation_type   2870 non-null   object
 5   cost                 2870 non-null   int64
 6   minimum_nights       2870 non-null   int64
 7   number_of_reviews    2870 non-null   int64
 8   reviews_per_month    2194 non-null   float64
 9   owner_id             2870 non-null   int64
 10  owned_hotels         2870 non-null   int64
 11  yearly_availability  2870 non-null   int64
dtypes: float64(3), int64(7), object(2)
```

Custom summary of numerical features.

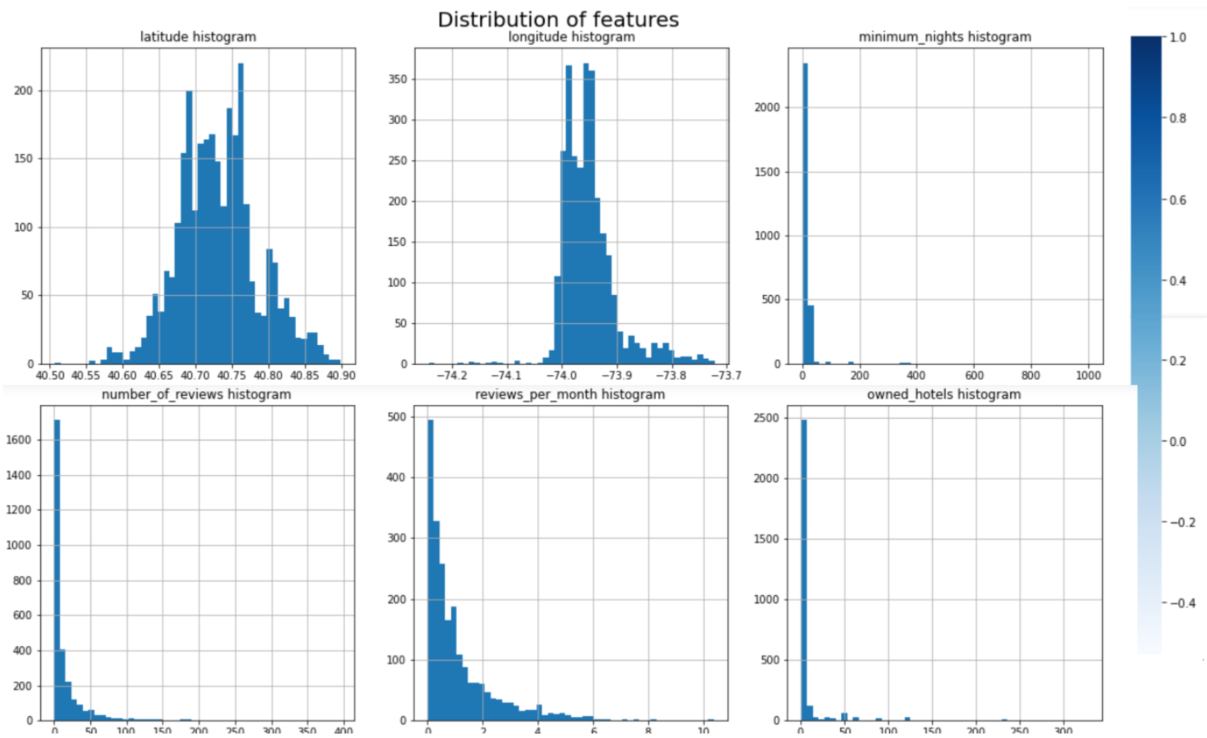
	Feature_name	datatype	Count	min	quartile1	Mean	Median	quartile3	max	Std dev	Skewness	Kurtosis	Range	IQR	skewness comment	outlier comment
0	latitude	float64	2870	40.50708	40.692462	40.731224	40.72825	40.762658	40.89873	0.05	0.17	0.21	0.39165	0.070195	Fairly symmetric(positive)	Has outliers
1	longitude	float64	2870	-74.24285	-73.984003	-73.950158	-73.95672	-73.934202	-73.72173	0.05	1.36	4.43	0.52112	0.049800	High positive skewed	Has outliers
2	cost	int64	2870	10.00000	75.000000	195.943206	120.00000	200.000000	9999.00000	406.18	13.01	232.35	9989.00000	125.000000	High positive skewed	Has outliers
3	minimum_nights	int64	2870	1.00000	1.000000	11.530314	3.00000	6.000000	999.00000	37.97	11.87	210.77	998.00000	5.000000	High positive skewed	Has outliers
4	number_of_reviews	int64	2870	0.00000	1.000000	16.315331	4.00000	16.000000	395.00000	32.48	4.27	25.44	395.00000	15.000000	High positive skewed	Has outliers
5	reviews_per_month	float64	2194	0.01000	0.240000	1.157502	0.65000	1.530000	10.37000	1.36	2.16	5.81	10.36000	1.290000	High positive skewed	Has outliers
6	owned_hotels	int64	2870	1.00000	1.000000	8.411498	1.00000	3.000000	327.00000	27.11	6.95	62.60	326.00000	2.000000	High positive skewed	Has outliers
7	yearly_availability	int64	2870	0.00000	0.000000	0.498606	0.00000	1.000000	1.00000	0.50	0.01	-2.00	1.00000	1.000000	Fairly symmetric(positive)	No outliers

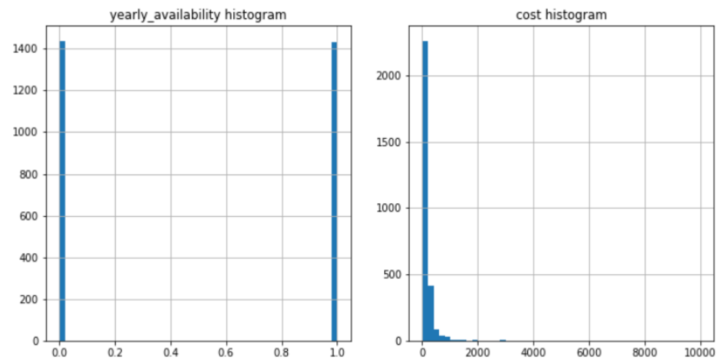
Variables id, owner_id have been dropped as they are unique to the customer and doesn't much value to the model's prediction.
Variables accommodation type and region being categorical variables, so they have been encoded using label encoding.

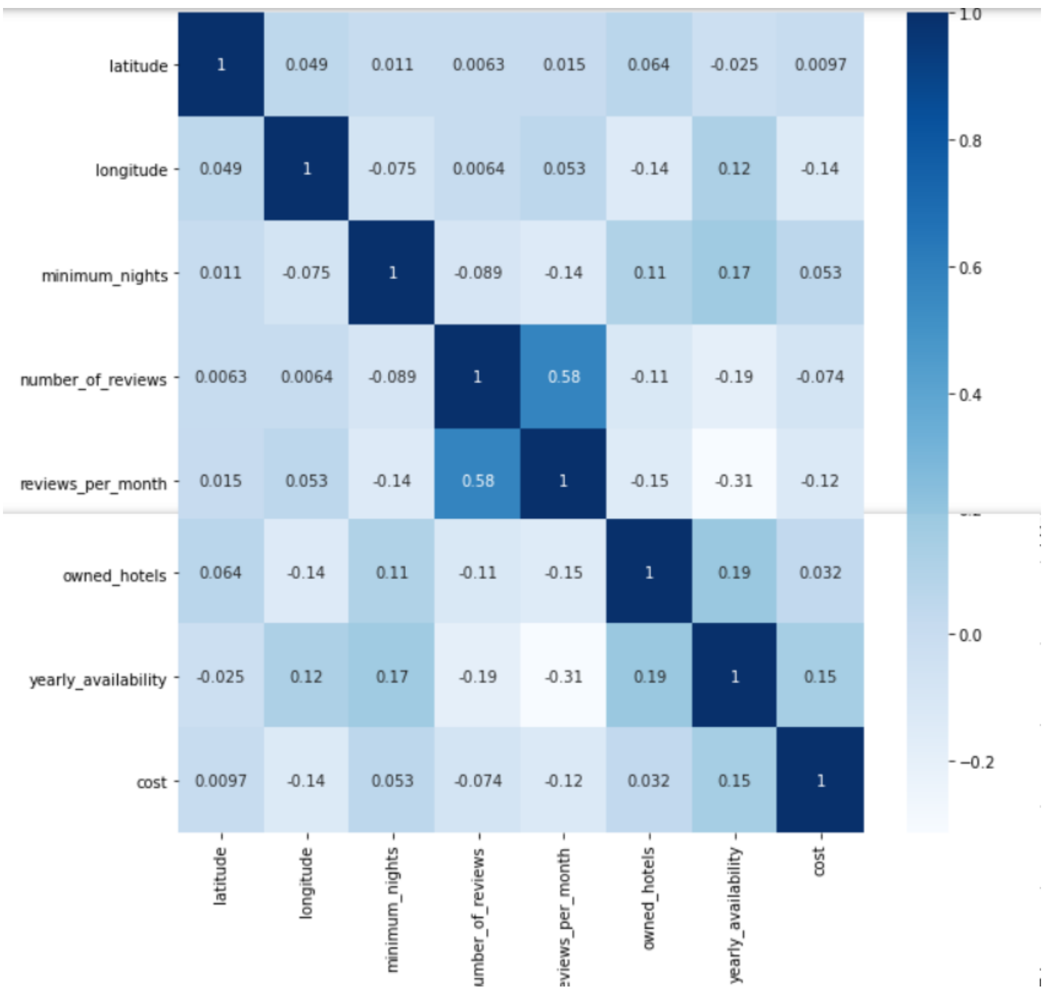
600+ Null values were detected in the review_per_month variables in the data and they have been imputed with zero because there were no reviews in the number_of_review column for those corresponding null values.

Response variable 'cost' are highly positively skewed. Which makes this regression exercise very difficult and even the R-squared and Adj R-squared values are very low.

Distribution of all features







Feature selection

From the three feature selection techniques forward, backward and stepwise, all the 8 independent variables were selected.

After the feature selection, scaling was performed on the independent datasets using standard scalar.

Model results.

************** R^2 of Models **************

Linear Regression: R^2 score is 0.0414320624453266
Linear Regression: Adjusted R^2 score is 0.03942318796913802

Best alpha 2.7304476164675275
Lasso Regression: R^2 score is 0.040555507468974406
Lasso Regression: Adjusted R^2 score is 0.03854479599318461

Best Alpha 0.9899999999999995
Ridge Regression: R^2 score is 0.04142602451455524
Ridge Regression: Adjusted R^2 score is 0.03941713738465202

Quadratic Regression: R^2 score is 0.05442672668346937
Quadratic Regression: Adjusted R^2 score is 0.052445085174597894

Symbolic Regression: R^2 score is -0.024828426607538034
Symbolic Regression: Adjusted R^2 score is -0.026976163442901324

************** R^2-cross-validated of Models **************
Linear Regression R^2 score is: 0.033 (0.044)

Lasso Regression R^2 score is: 0.038 (0.038)

Ridge Regression R^2 score is: 0.033 (0.044)

Quadratic Regression R^2 score is: 0.033 (0.044)

Symbolic Regression R^2 score is: -0.099 (0.169)

<u>Auto MPG:</u>

Data Set Information:

This dataset is a slightly modified version of the dataset provided in the StatLib library. In line with the use by Ross Quinlan (1993) in predicting the attribute "mpg", 8 of the original instances were removed because they had unknown values for the "mpg" attribute. The original dataset is available in the file "auto-mpg.data-original".

Attribute Information:

1. mpg: continuous

2. cylinders: multi-valued discrete

3. displacement: continuous

4. horsepower: continuous

5. weight: continuous

6. acceleration: continuous

7. model year: multi-valued discrete

8. origin: multi-valued discrete

9. car name: string (unique for each instance)

```
data.info()
data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    int64
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model year    398 non-null    int64
 7   origin        398 non-null    int64
 8   car name      398 non-null    int64
dtypes: float64(3), int64(6)
memory usage: 28.1 KB
```

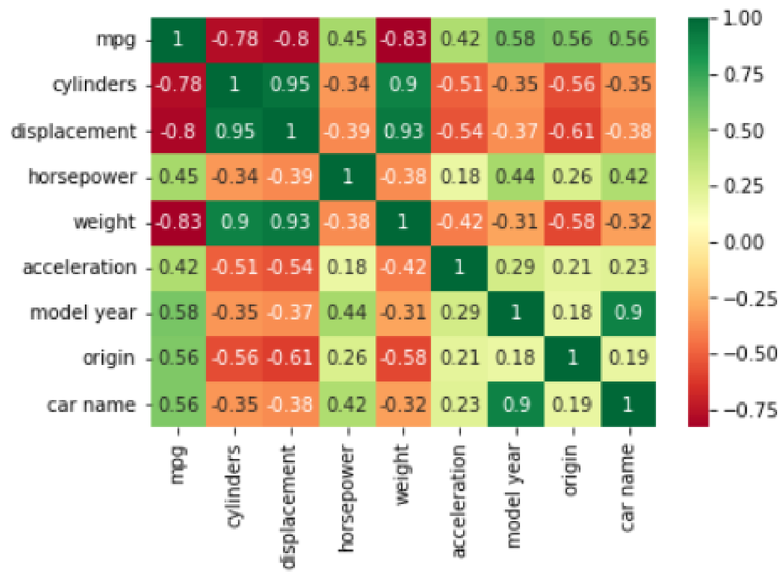
Exploratory Data Analysis

[5]:

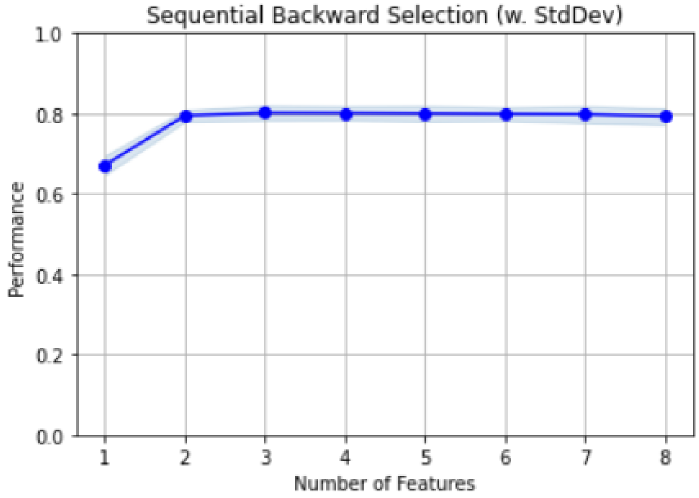
```
data.drop_duplicates()
print(data.size)
corr = data.corr()

sns.heatmap(corr, cmap ='RdYlGn', annot = True)
plt.show()
```

3582



```
2   0.007215
1   0.011868
```



In [13]:
```
print('\nSequential Backward Selection best:')
print(sfs2.k_feature_idx_)
print('CV Score:')
print(sfs2.k_score_)
```

```
Sequential Backward Selection best:
(3, 5, 6)
CV Score:
0.8016951001447611
```

Models:

Linear Regression:

R2: 0.8350774853140814

Adj R2: 0.8299236567301465

Lasso Regression

R2 0.810882236598091

Adj R2 0.8049723064917813

Ridge Regression

R2: 0.8351387384635716

Adj R2: 0.8299868240405581

Quadratic Regression

R2 0.8857505723869155

Adj R2 0.8821802777740065

Symbolic Regression
R2 0.8985500344084522
Adj R2 0.8953797229837163

Ridge Symbolic Regression
R2 0.5374527235645212
Adj R2 0.5302999306299521

Lasso Symbolic Regression
R2 0.7267487060204503
Adj R2 0.7225231705465397

Best Model Symbolic Regression Based on R2

FoldsCpp

The dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the power plant was set to work with full load. Features consist of hourly average ambient variables Temperature (T), Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (V) to predict the net hourly electrical energy output (EP) of the plant.
A combined cycle power plant (CCPP) is composed of gas turbines (GT), steam turbines (ST) and heat recovery steam generators. In a CCPP, the electricity is generated by gas and steam turbines, which are combined in one cycle, and is transferred from one turbine to another. While the Vacuum is colected from and has effect on the Steam Turbine, he other three of the ambient variables effect the GT performance.
For comparability with our baseline studies, and to allow 5x2 fold statistical tests be carried out, we provide the data shuffled five times. For each shuffling 2-fold CV is carried out and the resulting 10 measurements are used for statistical testing.

Data Overview:

```
In [28]: data.head()
```

Out[28]:

	AT	V	AP	RH	PE
0	8.34	40.77	1010.84	90.01	480.48
1	23.64	58.49	1011.40	74.20	445.75
2	29.74	56.90	1007.15	41.91	438.76
3	19.07	49.69	1007.22	76.79	453.09
4	11.80	40.66	1017.13	97.20	464.43

Exploratory Data Analysis

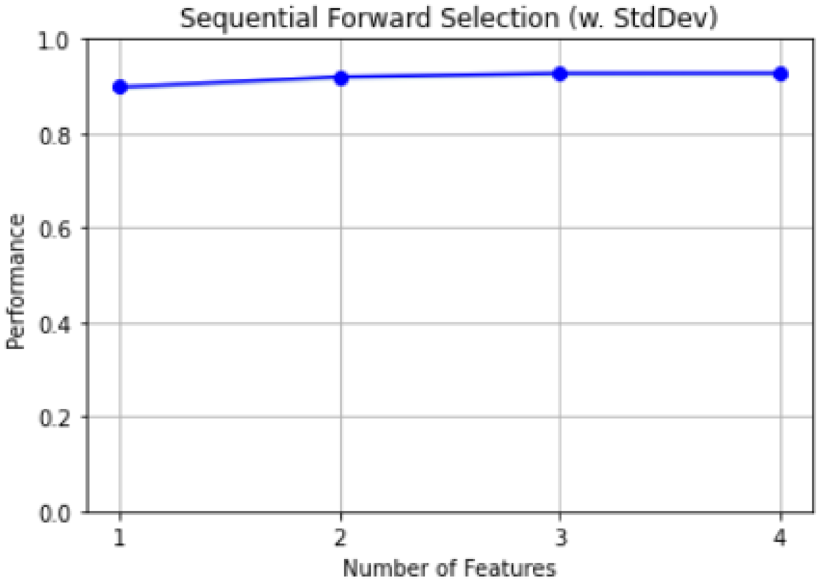
In [6]:
```python
data.drop_duplicates()
print(data.size)
corr = data.corr()


X = [ 'AT', 'V', 'AP', 'RH' ]
Y = ['PE']
sns.heatmap(corr, cmap ='RdYlGn', annot = True)
plt.show()
```

```
47840
```


```
Sequential Forward Selection best:
(0, 1, 2, 3)
CV Score:
0.9279901509689414
('AT', 'V', 'AP', 'RH')
```

	feature_names	ci_bound	std_dev	std_err
1	(AT,)	0.00339	0.002637	0.001319
2	(AT, RH)	0.004615	0.003591	0.001795
3	(AT, V, RH)	0.004988	0.003881	0.00194
4	(AT, V, AP, RH)	0.005055	0.003933	0.001966



Models:

Linear Regression:

R2: 0.9303315024744818

Adj R2: 0.9302147559348496

Lasso Regression

R2: 0.9303156097596659

Adj R2: 0.9301988365879184

Ridge Regression

Best Alpha 0.1

R2: 0.9303315032356659

Adj R2: 0.9302147566973092

Quadratic Regression

R2: 0.9388469766372023

Adj R2: 0.9387444998489949

Symbolic Regression

R2: 0.8973829722221228
Adj R2: 0.8972110123934209

Ridge Symbolic Regression
R2: 0.9363257970916933
Adj R2: 0.9362699792500029

Lasso Symbolic Regression
R2: 0.9304983804926836
Adj R2: 0.9304374542428416

Best Model Ridge Symbolic Regression **Based on R2**

USA_Housing Dataset:

We have taken this dataset from Kaggle. The main motive is to predict the price of the house. This dataset contains 6 attributes which are - Avg. Area Income, Avg. Area House Age,Avg. Area Number of Rooms,Avg. Area Number of Bedrooms,Area Population,Price.

Data Overview:

```
dataset.head()
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05

```
dataset.describe()
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562390	5.322283	6.299250	3.140000	29403.928700	9.975771e+05
50%	68804.286405	5.970429	7.002902	4.050000	36199.406690	1.232669e+06
75%	75783.338665	6.650808	7.665871	4.490000	42861.290770	1.471210e+06
max	107701.748400	9.519088	10.759588	6.500000	69621.713380	2.469066e+06

We have done the feature selection using Forward, Backward and Stepwise below:

Forward Feature Selection:

```
Selected Features: ('Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Area Population')
Selected Features ID: (0, 1, 2, 4)
```

Backward Feature Selection:

```
Selected Features: ('Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Area Population')
Selected Features ID: (0, 1, 2, 4)
```

Step-wise Feature Selection:

```
Selected Features: ('Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Area Population')
Selected Features ID: (0, 1, 2, 4)
```

Forward, backward and step-wise feature selections gave the same best features.

Summary of the Regressions:-

Linear Regression

R2 0.9248484536439604
Adj R2 1.0187644076005207

Ridge Regression

R2 0.9248481241195499
Adj R2 1.0187644898786727

Lasso Ridge

R2 0.5880657922589642
Adj R2 1.1028548546696564

Quadratic Regression

R2 0.9246788377522168
Adj R2 1.0188067585817384

Symbolic Regression

R2 0.8004152433298923
Adj R2 1.0498338345197296

Symbolic Ridge Regression

R2 0.917592566073667
Adj R2 1.0352838456028486

Symbolic Lasso Ridge

R2 0.917599121082713
Adj R2 1.0352810389880072

=> Linear Regression is an optimal model.