Appliances Energy Prediction using SVR

In [1]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.svm import SVR
from sklearn. metrics import r2_score, mean_absolute_error, mean_squared_error
import warnings
warnings.filterwarnings('ignore')
```

Reading dataset

In [2]:
```
df = pd.read_csv('data/energydata_complete.csv')

df.head()
```

Out[2]:

	date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	...	T9	RH_
0	2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.790000	19.79	44.730000	19.000000	...	17.033333	45.5
1	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	...	17.066667	45.5
2	2016-01-11 17:20:00	50	30	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	...	17.000000	45.5
3	2016-01-11 17:30:00	50	40	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	...	17.000000	45.4
4	2016-01-11 17:40:00	60	40	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	...	17.000000	45.4

5 rows × 29 columns

In [3]:
```
df.shape
```

Out[3]:
```
(19735, 29)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19735 entries, 0 to 19734
Data columns (total 29 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   date         19735 non-null  object
 1   Appliances   19735 non-null  int64
 2   lights       19735 non-null  int64
 3   T1           19735 non-null  float64
 4   RH_1         19735 non-null  float64
 5   T2           19735 non-null  float64
```

```
 6   RH_2         19735 non-null  float64
 7   T3           19735 non-null  float64
 8   RH_3         19735 non-null  float64
 9   T4           19735 non-null  float64
 10  RH_4         19735 non-null  float64
 11  T5           19735 non-null  float64
 12  RH_5         19735 non-null  float64
 13  T6           19735 non-null  float64
 14  RH_6         19735 non-null  float64
 15  T7           19735 non-null  float64
 16  RH_7         19735 non-null  float64
 17  T8           19735 non-null  float64
 18  RH_8         19735 non-null  float64
 19  T9           19735 non-null  float64
 20  RH_9         19735 non-null  float64
 21  T_out        19735 non-null  float64
 22  Press_mm_hg  19735 non-null  float64
 23  RH_out       19735 non-null  float64
 24  Windspeed    19735 non-null  float64
 25  Visibility   19735 non-null  float64
 26  Tdewpoint    19735 non-null  float64
 27  rv1          19735 non-null  float64
 28  rv2          19735 non-null  float64
dtypes: float64(26), int64(2), object(1)
memory usage: 4.4+ MB
```

In [5]:
```
df = df.set_index('date')
df.head()
```

Out[5]:

	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	...	
date												
2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.790000	19.79	44.730000	19.000000	45.566667	...	17.033
2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	45.992500	...	17.066
2016-01-11 17:20:00	50	30	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	45.890000	...	17.000
2016-01-11 17:30:00	50	40	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	45.723333	...	17.000
2016-01-11 17:40:00	60	40	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	45.530000	...	17.000

5 rows × 28 columns

EDA

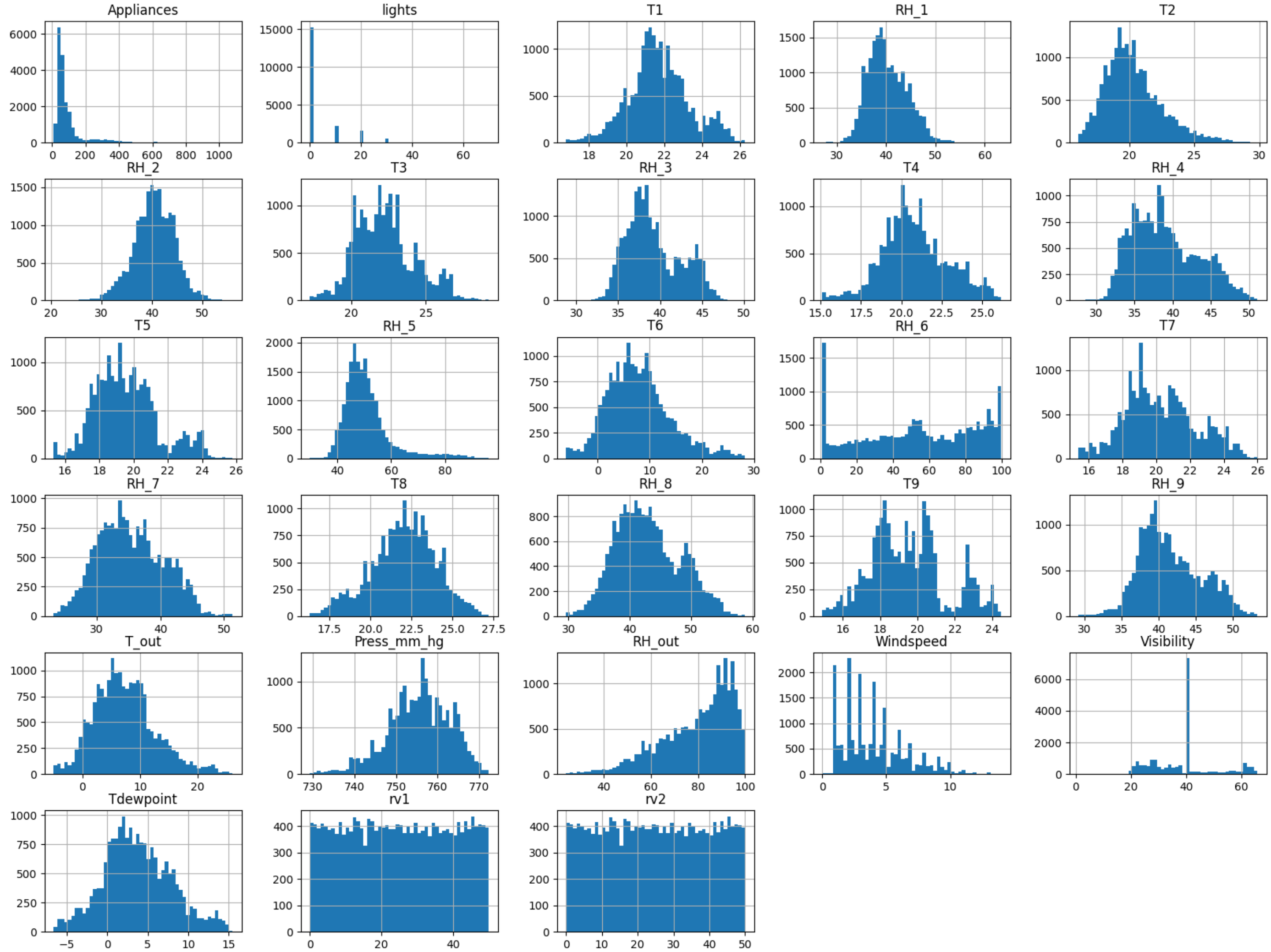
In [6]:
```
df.describe().T
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
Appliances	19735.0	97.694958	102.524891	10.000000	50.000000	60.000000	100.000000	1080.000000
lights	19735.0	3.801875	7.935988	0.000000	0.000000	0.000000	0.000000	70.000000

T1	19735.0	21.686571	1.606066	16.790000	20.760000	21.600000	22.600000	26.260000
RH_1	19735.0	40.259739	3.979299	27.023333	37.333333	39.656667	43.066667	63.360000
T2	19735.0	20.341219	2.192974	16.100000	18.790000	20.000000	21.500000	29.856667
RH_2	19735.0	40.420420	4.069813	20.463333	37.900000	40.500000	43.260000	56.026667
T3	19735.0	22.267611	2.006111	17.200000	20.790000	22.100000	23.290000	29.236000
RH_3	19735.0	39.242500	3.254576	28.766667	36.900000	38.530000	41.760000	50.163333
T4	19735.0	20.855335	2.042884	15.100000	19.530000	20.666667	22.100000	26.200000
RH_4	19735.0	39.026904	4.341321	27.660000	35.530000	38.400000	42.156667	51.090000
T5	19735.0	19.592106	1.844623	15.330000	18.277500	19.390000	20.619643	25.795000
RH_5	19735.0	50.949283	9.022034	29.815000	45.400000	49.090000	53.663333	96.321667
T6	19735.0	7.910939	6.090347	-6.065000	3.626667	7.300000	11.256000	28.290000
RH_6	19735.0	54.609083	31.149806	1.000000	30.025000	55.290000	83.226667	99.900000
T7	19735.0	20.267106	2.109993	15.390000	18.700000	20.033333	21.600000	26.000000
RH_7	19735.0	35.388200	5.114208	23.200000	31.500000	34.863333	39.000000	51.400000
T8	19735.0	22.029107	1.956162	16.306667	20.790000	22.100000	23.390000	27.230000
RH_8	19735.0	42.936165	5.224361	29.600000	39.066667	42.375000	46.536000	58.780000
T9	19735.0	19.485828	2.014712	14.890000	18.000000	19.390000	20.600000	24.500000
RH_9	19735.0	41.552401	4.151497	29.166667	38.500000	40.900000	44.338095	53.326667
T_out	19735.0	7.411665	5.317409	-5.000000	3.666667	6.916667	10.408333	26.100000
Press_mm_hg	19735.0	755.522602	7.399441	729.300000	750.933333	756.100000	760.933333	772.300000
RH_out	19735.0	79.750418	14.901088	24.000000	70.333333	83.666667	91.666667	100.000000
Windspeed	19735.0	4.039752	2.451221	0.000000	2.000000	3.666667	5.500000	14.000000
Visibility	19735.0	38.330834	11.794719	1.000000	29.000000	40.000000	40.000000	66.000000
Tdewpoint	19735.0	3.760707	4.194648	-6.600000	0.900000	3.433333	6.566667	15.500000
rv1	19735.0	24.988033	14.496634	0.005322	12.497889	24.897653	37.583769	49.996530
rv2	19735.0	24.988033	14.496634	0.005322	12.497889	24.897653	37.583769	49.996530

In [7]:
```
#plotting distributions of features
df.hist(bins=50, figsize=(20,15))
plt.show()
```



- Distribution of appliances is right skewed
- Most of the records in light column have value 0
- RH_out is left skewed
- Visibily has a irregular distribution, most number of datapoints have value close to 40
- All the temperature and most of the humidity columns follow normal distribution

In [8]:

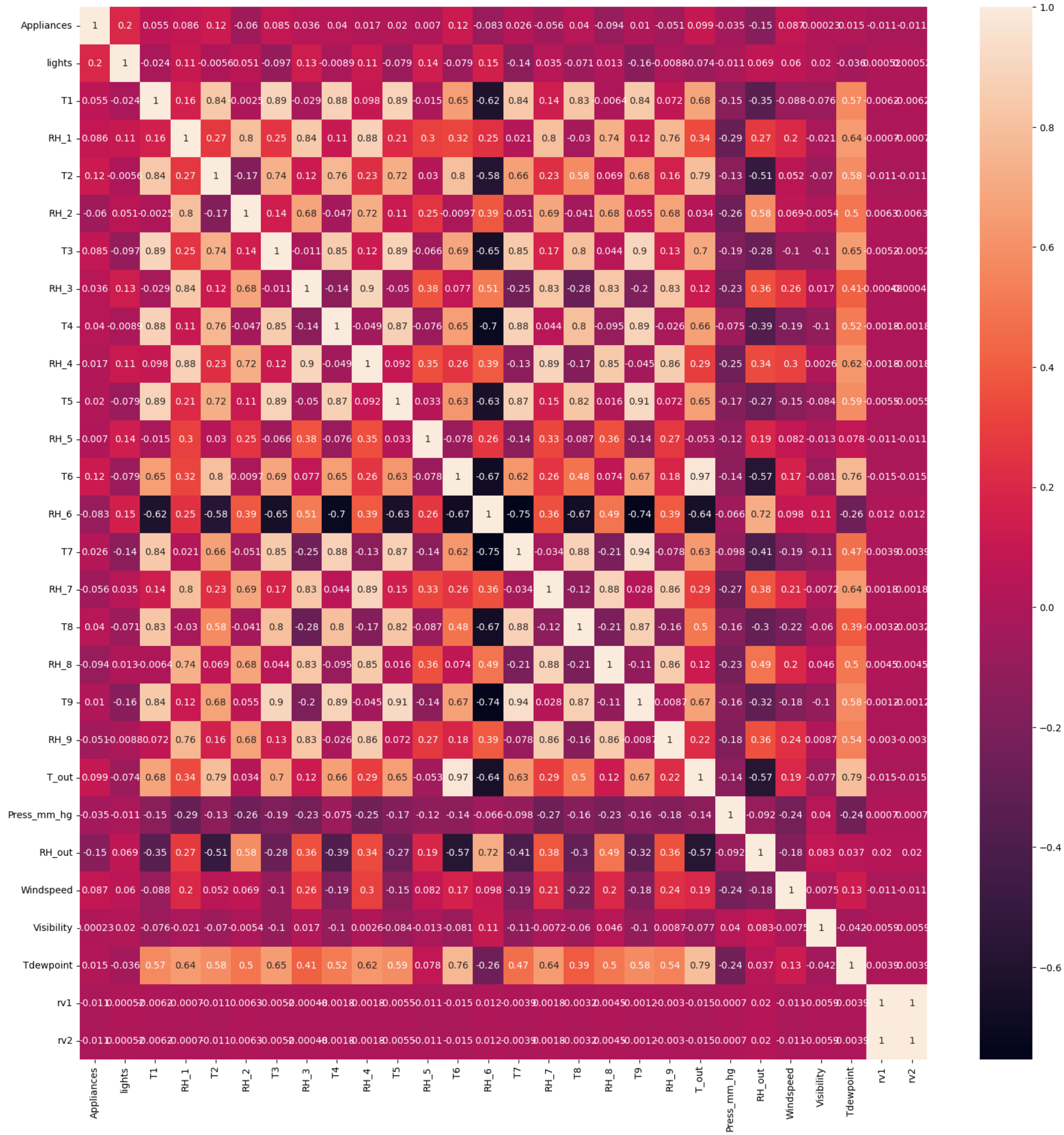
```
#correlation heatmap

corr = df.corr()
plt.figure(figsize=(20,20))

sns.heatmap(corr, annot=True)
```

Out[8]:

```
<AxesSubplot: >
```



- Other vairalbes doesn't show any strong positive correlation with targe Appliances
- Temperature columns have multicollinearity
- Humidity columns shows multicolllinearity
- columns rv1 and rv2 seems irrelevent

Data preprocessing

In [9]:
```
#checking for null values
df.isnull().sum()
```

Out[9]:
```
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
```

In [10]:
```
#checking for duplicate values
df.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
#creating fetature matrix and target vector
X = df.drop(['Appliances', 'T2', 'T3', 'T4', 'T5', 'T7', 'T8', 'T9', 'rv1', 'rv2'], axis
y = df['Appliances']
```

In [12]:
```
#splitting dataset into train and test splits
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
```

In [13]:
```
#feature scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler = scaler.fit(X_train)

X_trainScl = scaler.transform(X_train)
X_testScl = scaler.transform(X_test)
```

Model Building

In [22]:
```
regressor = SVR(kernel='rbf', C=10000)
regressor.fit(X_trainScl, y_train)
```

Out[22]:
```
▾ SVR
SVR(C=10000)
```

In [23]:
```
y_pred = regressor.predict(X_testScl)

r2_score(y_test, y_pred)
```

Out[23]:
```
0.39455366162908423
```