In [3]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [4]:

```
dataset = pd.read_csv('/Users/Sunil Gupta/Desktop/projects/USAR ml project/UCI CBM datase
dataset.head()
```

Out[4]:

	1 - Lever position (lp) [ ]	2 - Ship speed (v) [knots]	3 - Gas Turbine shaft torque (GTT) [kN m]	4 - Gas Turbine rate of revolutions (GTn) [rpm]	5 - Gas Generator rate of revolutions (GGn) [rpm]	6 - Starboard Propeller Torque (Ts) [kN]	7 - Port Propeller Torque (Tp) [kN]	8 - HP Turbine exit temperature (T48) [C]	Con tem
0	2.09	6.0	6960.0	1380.0	6830.0	28.2	28.2	635.0	
1	3.14	9.0	8380.0	1390.0	7110.0	60.4	60.4	606.0	
2	4.16	12.0	14700.0	1550.0	7790.0	114.0	114.0	661.0	
3	5.14	15.0	21600.0	1920.0	8490.0	175.0	175.0	731.0	
4	6.18	18.0	29800.0	2310.0	8830.0	246.0	246.0	800.0	

In [5]:

```
dataset.size
```

Out[5]:

```
214794
```

In [6]:

```
dataset.shape
```

Out[6]:

```
(11933, 18)
```

In [8]:

```
dataset.values
```

Out[8]:

```
array([[2.09e+00, 6.00e+00, 6.96e+03, ..., 2.87e-01, 9.50e-01, 9.75e-01],
       [3.14e+00, 9.00e+00, 8.38e+03, ..., 2.59e-01, 9.50e-01, 9.75e-01],
       [4.16e+00, 1.20e+01, 1.47e+04, ..., 3.58e-01, 9.50e-01, 9.75e-01],
       ...,
       [7.15e+00, 2.10e+01, 3.90e+04, ..., 8.34e-01, 1.00e+00, 1.00e+00],
       [8.21e+00, 2.40e+01, 5.10e+04, ..., 1.15e+00, 1.00e+00, 1.00e+00],
       [9.30e+00, 2.70e+01, 7.28e+04, ..., 1.70e+00, 1.00e+00, 1.00e+00]])
```

In [9]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11933 entries, 0 to 11932
Data columns (total 18 columns):
 #   Column                                            Non-Null Count  D
type
---  ------                                            --------------  -
----
 0   1 - Lever position (lp) [ ]                       11933 non-null  f
loat64
 1   2 - Ship speed (v) [knots]                        11933 non-null  f
loat64
 2   3 - Gas Turbine shaft torque (GTT) [kN m]         11933 non-null  f
loat64
 3   4 - Gas Turbine rate of revolutions (GTn) [rpm]   11933 non-null  f
loat64
 4   5 - Gas Generator rate of revolutions (GGn) [rpm] 11933 non-null  f
loat64
 5   6 - Starboard Propeller Torque (Ts) [kN]          11933 non-null  f
loat64
 6   7 - Port Propeller Torque (Tp) [kN]               11933 non-null  f
loat64
 7   8 - HP Turbine exit temperature (T48) [C]         11933 non-null  f
loat64
 8   9 - GT Compressor inlet air temperature (T1) [C]  11933 non-null  f
loat64
 9   10 - GT Compressor outlet air temperature (T2) [C] 11933 non-null  f
loat64
 10  11 - HP Turbine exit pressure (P48) [bar]         11933 non-null  f
loat64
 11  12 - GT Compressor inlet air pressure (P1) [bar]  11933 non-null  f
loat64
 12  13 - GT Compressor outlet air pressure (P2) [bar] 11933 non-null  f
loat64
 13  14 - Gas Turbine exhaust gas pressure (Pexh) [bar] 11933 non-null  f
loat64
 14  15 - Turbine Injecton Control (TIC) [%]           11933 non-null  f
loat64
 15  16 - Fuel flow (mf) [kg/s]                        11933 non-null  f
loat64
 16  17 - GT Compressor decay state coefficient.       11933 non-null  f
loat64
 17  18 - GT Turbine decay state coefficient.          11933 non-null  f
loat64
dtypes: float64(18)
memory usage: 1.6 MB
```

In [10]:

```
dataset.isnull().sum()
```

Out[10]:

```
1 - Lever position (lp) [ ]                          0
2 - Ship speed (v) [knots]                           0
3 - Gas Turbine shaft torque (GTT) [kN m]            0
4 - Gas Turbine rate of revolutions (GTn) [rpm]      0
5 - Gas Generator rate of revolutions (GGn) [rpm]    0
6 - Starboard Propeller Torque (Ts) [kN]             0
7 - Port Propeller Torque (Tp) [kN]                  0
8 - HP Turbine exit temperature (T48) [C]            0
9 - GT Compressor inlet air temperature (T1) [C]     0
10 - GT Compressor outlet air temperature (T2) [C]   0
11 - HP Turbine exit pressure (P48) [bar]            0
12 - GT Compressor inlet air pressure (P1) [bar]     0
13 - GT Compressor outlet air pressure (P2) [bar]    0
14 - Gas Turbine exhaust gas pressure (Pexh) [bar]   0
15 - Turbine Injecton Control (TIC) [%]              0
16 - Fuel flow (mf) [kg/s]                           0
17 - GT Compressor decay state coefficient.          0
18 - GT Turbine decay state coefficient.             0
dtype: int64
```

In [12]:

```
dataset = dataset.dropna()
dataset.isna().sum()
```

Out[12]:

```
1 - Lever position (lp) [ ]                          0
2 - Ship speed (v) [knots]                           0
3 - Gas Turbine shaft torque (GTT) [kN m]            0
4 - Gas Turbine rate of revolutions (GTn) [rpm]      0
5 - Gas Generator rate of revolutions (GGn) [rpm]    0
6 - Starboard Propeller Torque (Ts) [kN]             0
7 - Port Propeller Torque (Tp) [kN]                  0
8 - HP Turbine exit temperature (T48) [C]            0
9 - GT Compressor inlet air temperature (T1) [C]     0
10 - GT Compressor outlet air temperature (T2) [C]   0
11 - HP Turbine exit pressure (P48) [bar]            0
12 - GT Compressor inlet air pressure (P1) [bar]     0
13 - GT Compressor outlet air pressure (P2) [bar]    0
14 - Gas Turbine exhaust gas pressure (Pexh) [bar]   0
15 - Turbine Injecton Control (TIC) [%]              0
16 - Fuel flow (mf) [kg/s]                           0
17 - GT Compressor decay state coefficient.          0
18 - GT Turbine decay state coefficient.             0
dtype: int64
```

In [71]:

```
def recognize_type(dataset, col, max_cat=20):
    if (dataset[col].dtype == "O") | (dataset[col].nunique() < max_cat):
        return "cat"
    else:
        return "num"


dic_cols = {col:recognize_type(dataset, col, max_cat=1) for col in dataset.columns}
print(dic_cols)
```

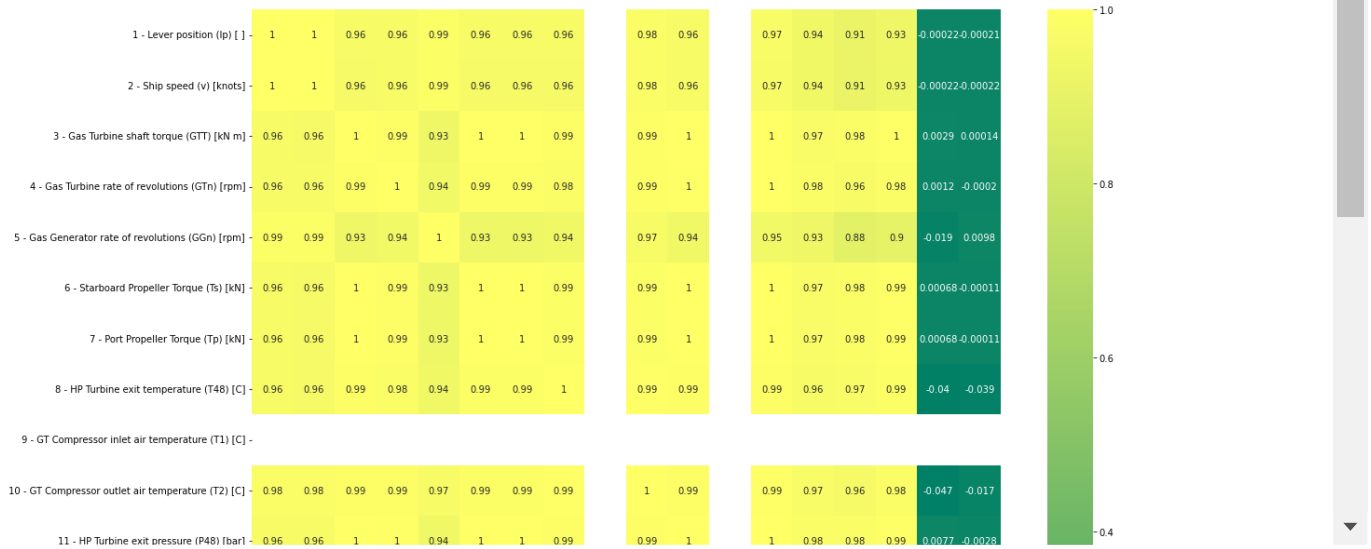
```
{'1 - Lever position (lp) [ ]': 'num', '2 - Ship speed (v) [knots]': 'nu
m', '3 - Gas Turbine shaft torque (GTT) [kN m]': 'num', '4 - Gas Turbine r
ate of revolutions (GTn) [rpm]': 'num', '5 - Gas Generator rate of revolut
ions (GGn) [rpm]': 'num', '6 - Starboard Propeller Torque (Ts) [kN]': 'nu
m', '7 - Port Propeller Torque (Tp) [kN]': 'num', '8 - HP Turbine exit tem
perature (T48) [C]': 'num', '9 - GT Compressor inlet air temperature (T1)
[C]': 'num', '10 - GT Compressor outlet air temperature (T2) [C]': 'num',
'11 - HP Turbine exit pressure (P48) [bar]': 'num', '12 - GT Compressor in
let air pressure (P1) [bar]': 'num', '13 - GT Compressor outlet air pressu
re (P2) [bar]': 'num', '14 - Gas Turbine exhaust gas pressure (Pexh) [ba
r]': 'num', '15 - Turbine Injecton Control (TIC) [%]': 'num', '16 - Fuel f
low (mf) [kg/s]': 'num', '17 - GT Compressor decay state coefficient.': 'n
um', '18 - GT Turbine decay state coefficient.': 'num'}
```

In [72]:

```
plt.figure(figsize=(18, 18))
sns.heatmap(dataset.corr(),cbar=True, annot=True ,cmap = 'summer')
```

Out[72]:

<AxesSubplot:>



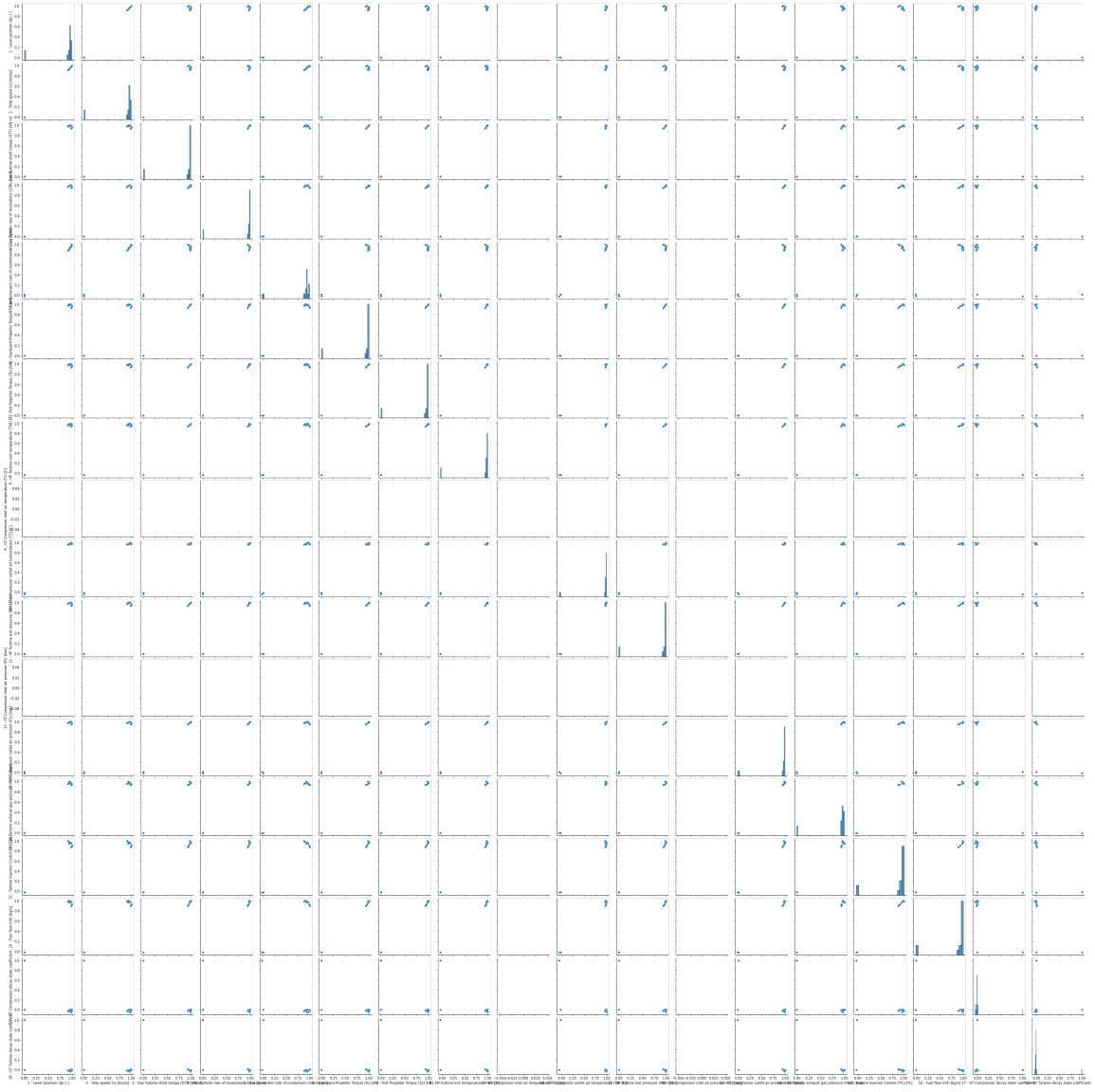
In [36]:

```
plt.figure(figsize=(18, 18))
sns.pairplot(dataset.corr())
```

Out[36]:

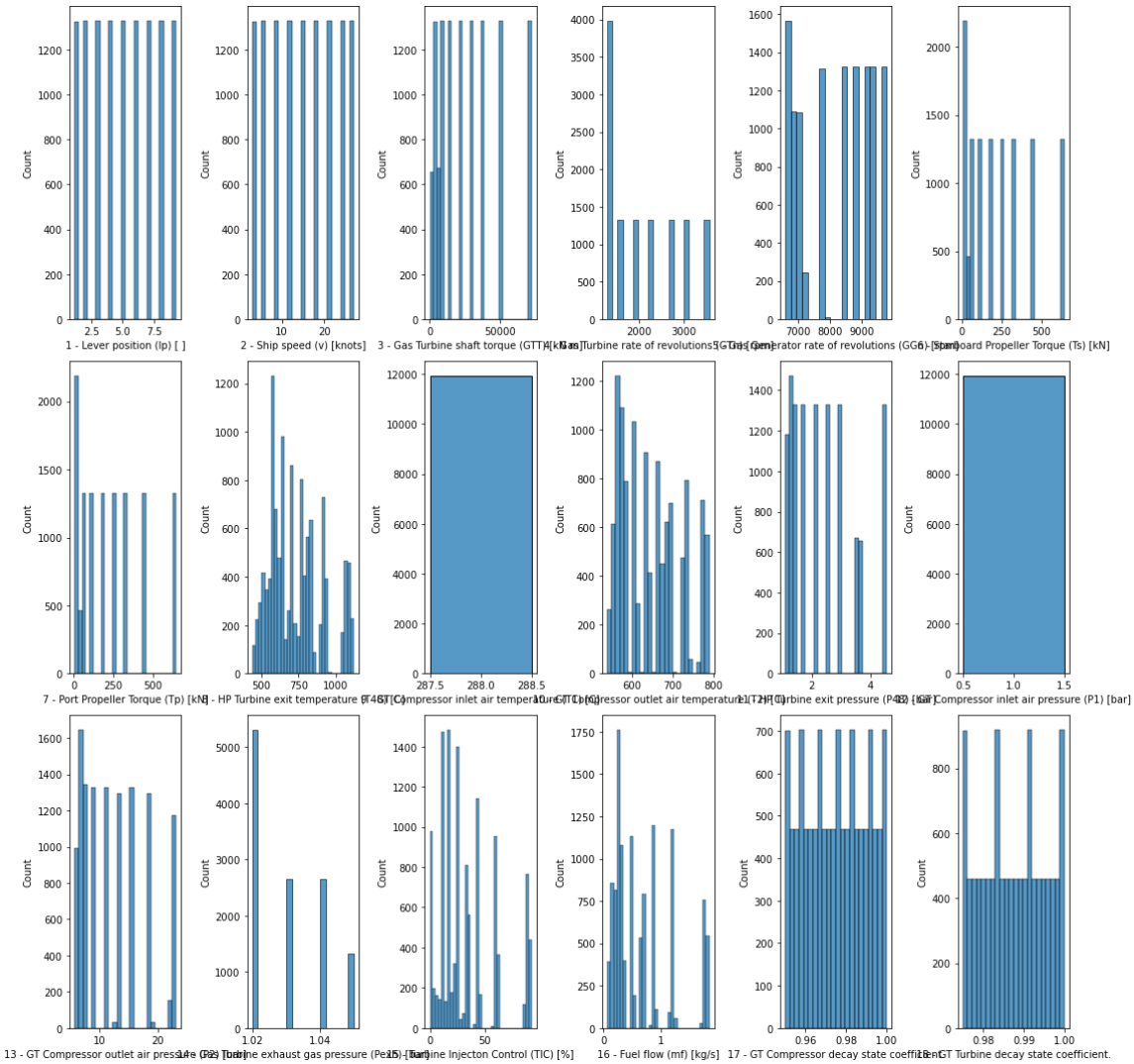
```
<seaborn.axisgrid.PairGrid at 0x1a0e2635460>
```

```
<Figure size 1296x1296 with 0 Axes>
```



In [40]:

```
count=1
plt.subplots(figsize=(15, 15))
for i in dataset.columns:
    plt.subplot(3,6,count)
    sns.histplot(dataset[i])
    count+=1

plt.tight_layout()
plt.show()
```



In [61]:

```
x = dataset.drop(['17 - GT Compressor decay state coefficient.','18 - GT Turbine decay st
y = dataset[['17 - GT Compressor decay state coefficient.','18 - GT Turbine decay state c
```

In [62]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,random_state=0)
```

In [63]:

```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(8353, 16)
(8353, 2)
(3580, 16)
(3580, 2)
```

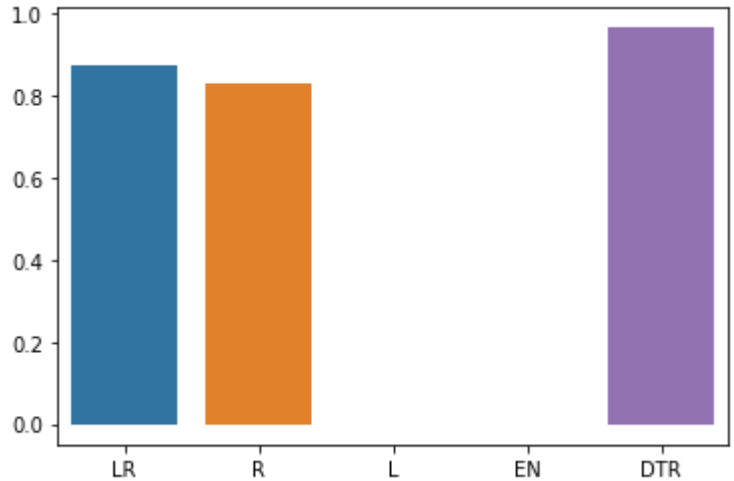
In [64]:

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge,Lasso,ElasticNet
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score

reg1 = LinearRegression()
reg2 = Ridge(alpha=1.0)
reg3 = Lasso()
reg4 = ElasticNet()
reg5 = DecisionTreeRegressor()

Reg = [reg1,reg2,reg3,reg4,reg5]
reg_name = ['LR','R','L','EN','DTR']
r_sq_score = {}

for model, model_name in zip(Reg, reg_name):
    model.fit(x_train,y_train)
    pred = model.predict(x_test)

    r_sq_score[model_name] = r2_score(y_test,pred)

print("R2_SCORES")
for i, j in r_sq_score.items():
    print(i, ':-', j)

print(r_sq_score.keys())
print(r_sq_score.values())

plt.figure()
sns.barplot(x=list(r_sq_score.keys()), y=list(r_sq_score.values()))
```

```
R2_SCORES
LR :- 0.873276199651367
R :- 0.8294719901685492
L :- -0.0007405519833493246
EN :- -0.0007496498732133539
DTR :- 0.9668534237127835
dict_keys(['LR', 'R', 'L', 'EN', 'DTR'])
dict_values([0.873276199651367, 0.8294719901685492, -0.0007405519833493246, -0.0007496498732133539, 0.9668534237127835])
```

Out[64]:

```
<AxesSubplot:>
```



In [65]:

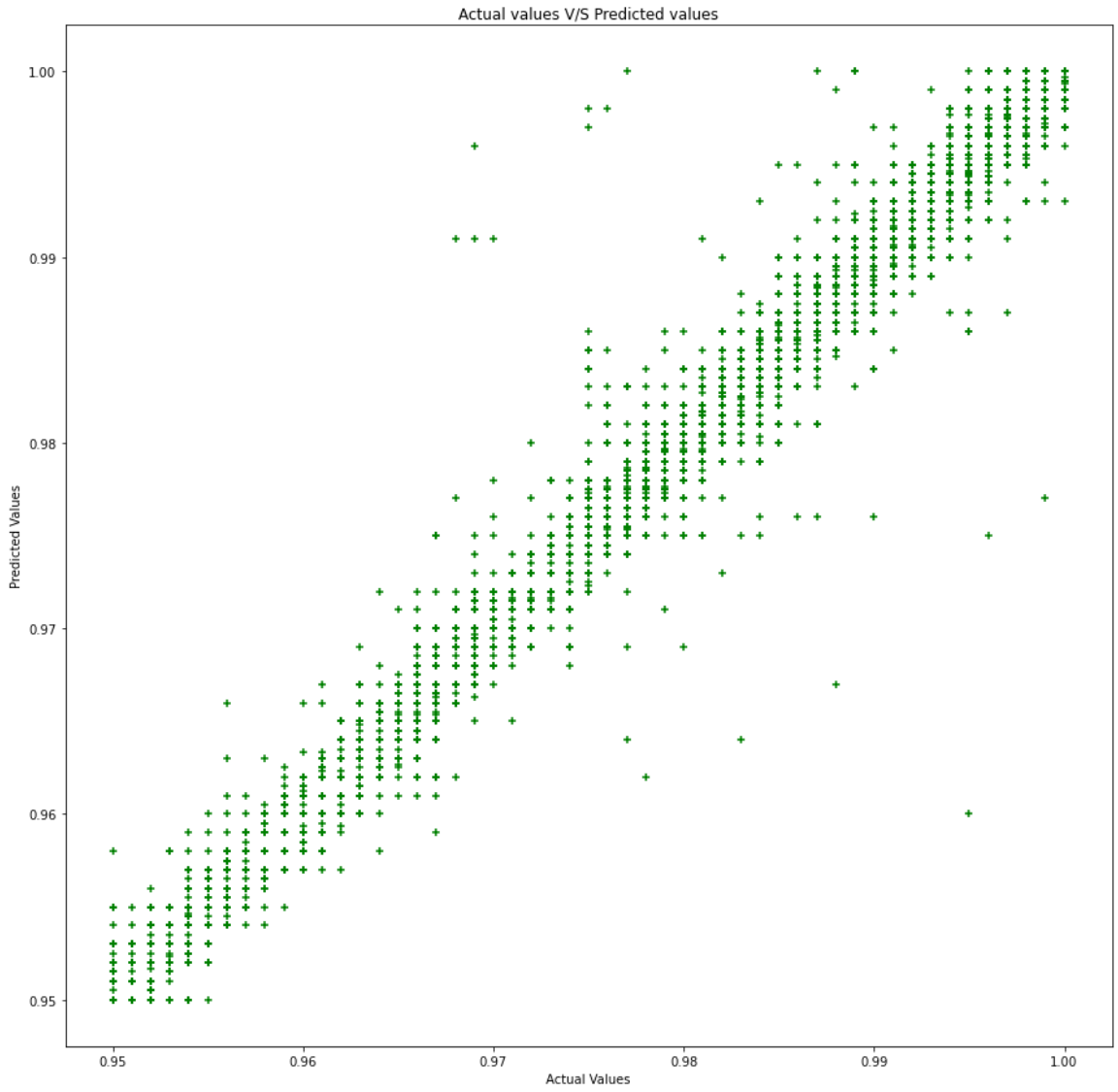
```
y_pred = reg5.predict(x_test)
```

In [66]:

```
plt.figure(figsize=(15,15))
plt.scatter(y_test,y_pred,color='green',marker='+')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual values V/S Predicted values')
```

Out[66]:

```
Text(0.5, 1.0, 'Actual values V/S Predicted values')
```



In [67]:

```
d = {'Actual':y_test,'Predicted':y_pred,'Difference':y_test - y_pred}
pred_y_df = pd.DataFrame([d])
pred_y_df
```

Out[67]:

	Actual	Predicted	Difference
0	[[0.987, 0.989], [0.972, 0.993], [0.971, 0.995...	[[0.981, 0.986], [0.97, 0.994], [0.971, 0.996]...	[[0.006000000000000005, 0.0030000000000000027]...

In [68]:

```
reg5.predict([[2.09E+00,6.00E+00,2.87E+03,1.36E+03,6.86E+03,2.55E+01,2.55E+01,5.00E+02,2.
]])
```

Out[68]:

```
array([[0.959, 1.   ]])
```

In [ ]: