Objective of the problem statement - predicting turbine energy yield (TEY) using ambient variables as features.

1. Import neccessary libraries

In [3]:
```
import pandas as pd
```

2. Import Data

In [4]:
```
Gas_turbines = pd.read_csv('gas_turbines.csv')
Gas_turbines
```

Out[4]:

	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
0	6.8594	1007.9	96.799	3.5000	19.663	1059.2	550.00	114.70	10.605	3.1547	82.722
1	6.7850	1008.4	97.118	3.4998	19.728	1059.3	550.00	114.72	10.598	3.2363	82.776
2	6.8977	1008.8	95.939	3.4824	19.779	1059.4	549.87	114.71	10.601	3.2012	82.468
3	7.0569	1009.2	95.249	3.4805	19.792	1059.6	549.99	114.72	10.606	3.1923	82.670
4	7.3978	1009.7	95.150	3.4976	19.765	1059.7	549.98	114.72	10.612	3.2484	82.311
...	...	...	...	...	...	...	...	...	...	...	...
15034	9.0301	1005.6	98.460	3.5421	19.164	1049.7	546.21	111.61	10.400	4.5186	79.559
15035	7.8879	1005.9	99.093	3.5059	19.414	1046.3	543.22	111.78	10.433	4.8470	79.917
15036	7.2647	1006.3	99.496	3.4770	19.530	1037.7	537.32	110.19	10.483	7.9632	90.912
15037	7.0060	1006.8	99.008	3.4486	19.377	1043.2	541.24	110.74	10.533	6.2494	93.227
15038	6.9279	1007.2	97.533	3.4275	19.306	1049.9	545.85	111.58	10.583	4.9816	92.498

15039 rows × 11 columns

3. Data understanding

3.1 Initial anlaysis

In [5]:
```
Gas_turbines .shape
```

Out[5]: (15039, 11)

In [6]:
```
Gas_turbines .isna().sum()
```

Out[6]:
```
AT      0
AP      0
AH      0
AFDP    0
GTEP    0
TIT     0
TAT     0
TEY     0
CDP     0
CO      0
NOX     0
dtype: int64
```

4. Data Preparation

In [7]:
```
Gas_turbines .head(30)
```

Out[7]:

	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
0	6.8594	1007.9	96.799	3.5000	19.663	1059.2	550.00	114.70	10.605	3.15470	82.722
1	6.7850	1008.4	97.118	3.4998	19.728	1059.3	550.00	114.72	10.598	3.23630	82.776
2	6.8977	1008.8	95.939	3.4824	19.779	1059.4	549.87	114.71	10.601	3.20120	82.468
3	7.0569	1009.2	95.249	3.4805	19.792	1059.6	549.99	114.72	10.606	3.19230	82.670
4	7.3978	1009.7	95.150	3.4976	19.765	1059.7	549.98	114.72	10.612	3.24840	82.311
5	7.6998	1010.7	92.708	3.5236	19.683	1059.8	549.97	114.72	10.626	3.44670	82.409
6	7.7901	1011.6	91.983	3.5298	19.659	1060.0	549.87	114.71	10.644	3.48740	82.440
7	7.7139	1012.7	91.348	3.5088	19.673	1059.8	549.92	114.71	10.656	3.60430	83.010
8	7.7975	1013.8	90.196	3.5141	19.634	1060.1	550.09	114.72	10.644	3.39430	82.284
9	8.0820	1015.0	88.597	4.0612	23.406	1083.0	550.21	131.70	11.679	1.90810	82.782
10	8.3047	1016.0	86.343	4.0870	23.747	1085.3	550.20	133.67	11.703	1.71180	81.995
11	8.4684	1016.1	86.491	4.0513	23.734	1085.1	550.14	134.24	11.775	1.46720	80.638
12	8.8856	1016.2	82.974	4.0503	23.869	1085.9	550.17	134.69	11.864	1.71130	80.533
13	9.3714	1016.6	79.980	4.0427	23.916	1085.9	550.09	134.68	11.860	1.66370	80.538
14	9.7962	1017.1	78.061	4.0613	23.962	1085.9	549.97	134.68	11.855	1.40610	80.463
15	9.6996	1017.9	76.382	4.0719	23.945	1085.9	549.84	134.69	11.850	1.40720	80.522
16	9.3111	1018.8	75.559	4.0578	23.890	1085.9	549.72	134.69	11.845	1.51930	80.648
17	8.6702	1019.8	77.476	4.0727	23.824	1085.9	549.60	134.69	11.840	1.42010	80.374
18	8.1375	1020.6	79.741	4.0432	23.832	1085.4	549.99	134.68	11.835	1.61560	80.639
19	7.0396	1021.2	83.102	4.0110	23.655	1084.9	550.03	134.68	11.830	1.69290	81.223
20	6.5345	1021.7	86.253	4.0678	23.801	1085.7	549.87	134.67	11.867	1.49750	84.556
21	6.4893	1022.2	86.190	4.1476	24.098	1087.6	549.66	133.78	12.014	1.34300	86.110
22	6.4935	1022.4	84.820	4.1916	24.053	1088.1	550.03	134.11	11.908	1.30340	86.607
23	6.0790	1022.8	85.053	3.5210	19.259	1057.0	548.09	112.80	10.565	4.64580	105.150
24	5.4134	1023.8	84.432	3.8312	21.433	1070.4	548.85	122.33	11.186	4.11650	105.730
25	5.4755	1024.0	82.830	4.1239	23.893	1087.2	549.96	133.94	11.968	1.46050	87.298
26	5.9958	1024.2	82.376	4.6570	28.128	1099.7	543.28	150.33	12.935	0.78039	82.101
27	6.7455	1024.3	80.521	4.0011	23.628	1085.0	550.06	134.24	11.913	1.43270	84.386
28	7.9304	1023.7	76.271	4.6570	28.433	1096.2	540.40	150.02	13.055	1.12220	75.403
29	9.3818	1022.8	56.158	4.6340	28.221	1100.1	543.62	150.50	12.971	0.91143	83.766

In [8]:
```
Gas_turbines .describe()
```

Out[8]:

	AT	AP	AH	AFDP	GTEP	TIT	
count	15039.000000	15039.00000	15039.000000	15039.000000	15039.000000	15039.000000	15039.(
mean	17.764381	1013.19924	79.124174	4.200294	25.419061	1083.798770	545.3
std	7.574323	6.41076	13.793439	0.760197	4.173916	16.527806	7.8
min	0.522300	985.85000	30.344000	2.087400	17.878000	1000.800000	512.4
25%	11.408000	1008.90000	69.750000	3.723900	23.294000	1079.600000	542.1
50%	18.186000	1012.80000	82.266000	4.186200	25.082000	1088.700000	549.8
75%	23.862500	1016.90000	90.043500	4.550900	27.184000	1096.000000	550.(
max	34.929000	1034.20000	100.200000	7.610600	37.402000	1100.800000	550.6

5. Model Building

5.1 Input output separation

In [9]:
```
X = Gas_turbines.drop(labels='TEY',axis=1,)
y = Gas_turbines[['TEY']]
```

In [10]:
```
X
```

Out[10]:

	AT	AP	AH	AFDP	GTEP	TIT	TAT	CDP	CO	NOX
0	6.8594	1007.9	96.799	3.5000	19.663	1059.2	550.00	10.605	3.1547	82.722
1	6.7850	1008.4	97.118	3.4998	19.728	1059.3	550.00	10.598	3.2363	82.776
2	6.8977	1008.8	95.939	3.4824	19.779	1059.4	549.87	10.601	3.2012	82.468
3	7.0569	1009.2	95.249	3.4805	19.792	1059.6	549.99	10.606	3.1923	82.670
4	7.3978	1009.7	95.150	3.4976	19.765	1059.7	549.98	10.612	3.2484	82.311
...	...	...	...	...	...	...	...	...	...	...
15034	9.0301	1005.6	98.460	3.5421	19.164	1049.7	546.21	10.400	4.5186	79.559
15035	7.8879	1005.9	99.093	3.5059	19.414	1046.3	543.22	10.433	4.8470	79.917
15036	7.2647	1006.3	99.496	3.4770	19.530	1037.7	537.32	10.483	7.9632	90.912
15037	7.0060	1006.8	99.008	3.4486	19.377	1043.2	541.24	10.533	6.2494	93.227
15038	6.9279	1007.2	97.533	3.4275	19.306	1049.9	545.85	10.583	4.9816	92.498

15039 rows × 10 columns

In [11]:
```
y
```

Out[11]:

	TEY
0	114.70
1	114.72
2	114.71
3	114.72
4	114.72
...	...
15034	111.61
15035	111.78
15036	110.19
15037	110.74
15038	111.58

15039 rows × 1 columns

In [12]:
```
from sklearn.preprocessing import StandardScaler
Scaler = StandardScaler()
```

In [13]:
```
Scaler.fit(X)
```

Out[13]: StandardScaler()

In [14]:
```
X
```

Out[14]:

	AT	AP	AH	AFDP	GTEP	TIT	TAT	CDP	CO	NOX
0	6.8594	1007.9	96.799	3.5000	19.663	1059.2	550.00	10.605	3.1547	82.722
1	6.7850	1008.4	97.118	3.4998	19.728	1059.3	550.00	10.598	3.2363	82.776
2	6.8977	1008.8	95.939	3.4824	19.779	1059.4	549.87	10.601	3.2012	82.468
3	7.0569	1009.2	95.249	3.4805	19.792	1059.6	549.99	10.606	3.1923	82.670
4	7.3978	1009.7	95.150	3.4976	19.765	1059.7	549.98	10.612	3.2484	82.311
...	...	...	...	...	...	...	...	...	...	...
15034	9.0301	1005.6	98.460	3.5421	19.164	1049.7	546.21	10.400	4.5186	79.559
15035	7.8879	1005.9	99.093	3.5059	19.414	1046.3	543.22	10.433	4.8470	79.917
15036	7.2647	1006.3	99.496	3.4770	19.530	1037.7	537.32	10.483	7.9632	90.912
15037	7.0060	1006.8	99.008	3.4486	19.377	1043.2	541.24	10.533	6.2494	93.227
15038	6.9279	1007.2	97.533	3.4275	19.306	1049.9	545.85	10.583	4.9816	92.498

15039 rows × 10 columns

In [15]:
```
Gas_turbines .describe()
```

Out[15]:

	AT	AP	AH	AFDP	GTEP	TIT	
count	15039.000000	15039.00000	15039.000000	15039.000000	15039.000000	15039.000000	15039.(
mean	17.764381	1013.19924	79.124174	4.200294	25.419061	1083.798770	545.3
std	7.574323	6.41076	13.793439	0.760197	4.173916	16.527806	7.8
min	0.522300	985.85000	30.344000	2.087400	17.878000	1000.800000	512.4
25%	11.408000	1008.90000	69.750000	3.723900	23.294000	1079.600000	542.1
50%	18.186000	1012.80000	82.266000	4.186200	25.082000	1088.700000	549.8
75%	23.862500	1016.90000	90.043500	4.550900	27.184000	1096.000000	550.(
max	34.929000	1034.20000	100.200000	7.610600	37.402000	1100.800000	550.6

In [16]:
```
Gas_turbines.mean()
```

Out[16]:
```
AT         17.764381
AP       1013.199240
AH         79.124174
AFDP        4.200294
GTEP       25.419061
TIT      1083.798770
TAT       545.396183
TEY       134.188464
CDP        12.102353
CO          1.972499
NOX        68.190934
dtype: float64
```

In [17]:
```
Gas_turbines.std()
```

Out[17]:
```
AT        7.574323
AP        6.410760
AH       13.793439
AFDP      0.760197
GTEP      4.173916
TIT      16.527806
TAT       7.866803
TEY      15.829717
CDP       1.103196
CO        2.222206
NOX      10.470586
dtype: float64
```

5.2 Train test split

In [18]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size = 0.20, random_st
```

In [19]:
```
X_train
```

Out[19]:

	AT	AP	AH	AFDP	GTEP	TIT	TAT	CDP	CO	NOX
7606	9.5168	1023.2	92.122	2.4142	19.487	1041.9	539.39	10.522	10.48300	91.143
8322	6.8620	1019.6	77.369	5.1839	33.897	1100.0	524.08	14.307	0.28809	64.692
14243	10.9340	1026.5	88.000	3.9676	23.641	1085.2	549.90	11.831	1.70260	75.592
14326	12.4950	1020.7	80.112	5.2121	32.036	1100.1	530.57	13.909	0.60812	66.689
4594	24.6840	1017.2	73.637	3.4777	20.365	1060.6	549.93	10.772	2.93540	49.550
...	...	...	...	...	...	...	...	...	...	...
905	4.0458	1021.7	80.988	5.0520	28.201	1088.9	537.50	12.936	1.35100	79.469
5192	22.6190	1008.4	78.229	4.2732	25.779	1091.1	550.17	12.166	2.18120	67.182
12172	30.8550	1006.4	54.665	3.3196	21.010	1064.6	550.13	10.933	1.16880	48.937
235	7.3472	1019.3	77.571	5.0452	31.438	1097.2	530.47	13.823	1.06910	70.820
13349	21.6560	1013.9	91.497	4.3916	27.970	1094.7	544.10	12.755	1.16210	59.246

12031 rows × 10 columns

In [20]:
```
y_train
```

Out[20]:

	TEY
7606	110.21
8322	167.33
14243	133.73
14326	159.15
4594	110.78
...	...
905	147.95
5192	135.10
12172	112.86
235	159.09
13349	140.81

12031 rows × 1 columns

In [21]:
```
# For train data
X_train.shape,y_train.shape
```

Out[21]:
```
((12031, 10), (12031, 1))
```

In [22]:
```
# For train data
X_test.shape,y_test.shape
```

Out[22]:
```
((3008, 10), (3008, 1))
```

6. Model Training

In [28]:
```
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense, Activation,Layer,Lambda
```

In [29]:
```
model = Sequential()
model.add(Dense(units=20,activation='relu'))
model.add(Dense(units=40,activation='tanh'))
model.add(Dense(units=10,activation='softmax'))
```

In [30]:
```
model.compile(optimizer='adam',loss='mean_squared_error',metrics=['mse'])
```

7. Model Testing

In [31]:
```
model.fit(X_train,y_train,epochs=14)
```

```
Epoch 1/14
376/376 [==============================] - 3s 3ms/step - loss: 18241.0605 - m
se: 18241.0605
Epoch 2/14
376/376 [==============================] - 1s 3ms/step - loss: 18241.0625 - m
se: 18241.0625
Epoch 3/14
376/376 [==============================] - 1s 3ms/step - loss: 18241.0605 - m
se: 18241.0605
Epoch 4/14
376/376 [==============================] - 1s 3ms/step - loss: 18241.0547 - m
se: 18241.0547
Epoch 5/14
376/376 [==============================] - 1s 3ms/step - loss: 18241.0645 - m
se: 18241.0645
Epoch 6/14
376/376 [==============================] - 2s 5ms/step - loss: 18241.0547 - m
se: 18241.0547
Epoch 7/14
```

In [27]:
```
score = model.evaluate(X_test,y_test)
```

```
94/94 [==============================] - 1s 5ms/step - loss: 18187.1562 - mse:
18187.1562
```

8. Model Evaluation

In [41]:
```
result = model.evaluate(x=X_test,y=y_test)
result
```

```
94/94 [==============================] - 0s 2ms/step - loss: 18187.1562 - mse:
18187.1562
```

Out[41]:
```
[18187.15625, 18187.15625]
```

In [46]:
```
print('MSE : ',round(result[1],2))
print('Loss           : ',round(result[0],2))
```

```
MSE :  18187.16
Loss           :  18187.16
```

9. Model Deployment

In [43]:
```
model.save('Turbine_Energy.h5')
```