In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from keras.wrappers.scikit_learn import KerasRegressor
from keras.layers import Dense,Dropout
from sklearn.metrics import accuracy_score,mean_absolute_error,mean_squared_error
from sklearn.model_selection import GridSearchCV,KFold
from keras.models import Sequential
from tensorflow.keras.optimizers import Adam,RMSprop
```

In [2]:

```python
turbine_data = pd.read_csv('gas_turbines.csv')
turbine_data
```

Out[2]:

	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

In [3]:

```python
turbine_data.shape
```

Out[3]:

```
(15039, 11)
```

In [4]:
```
turbine_data.isna().sum()
```

Out[4]:
```
AT      0
AP      0
AH      0
AFDP    0
GTEP    0
TIT     0
TAT     0
TEY     0
CDP     0
CO      0
NOX     0
dtype: int64
```

In [5]:
```
turbine_data[turbine_data.duplicated()]
```

Out[5]:

AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX

In [6]:
```
turbine_data.dtypes
```

Out[6]:
```
AT      float64
AP      float64
AH      float64
AFDP    float64
GTEP    float64
TIT     float64
TAT     float64
TEY     float64
CDP     float64
CO      float64
NOX     float64
dtype: object
```

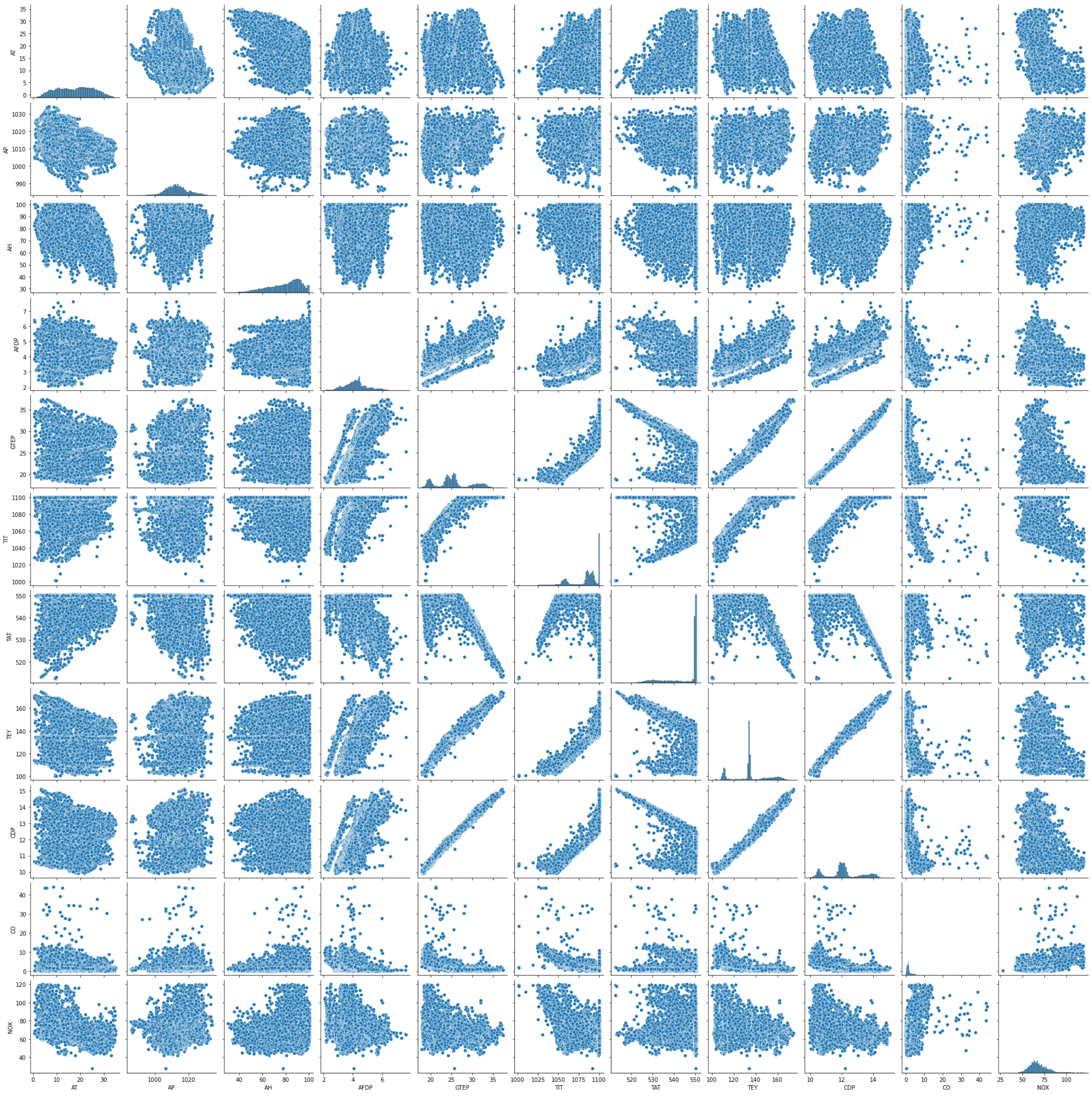
In [7]:
```
turbine_data.describe()
```

Out[7]:

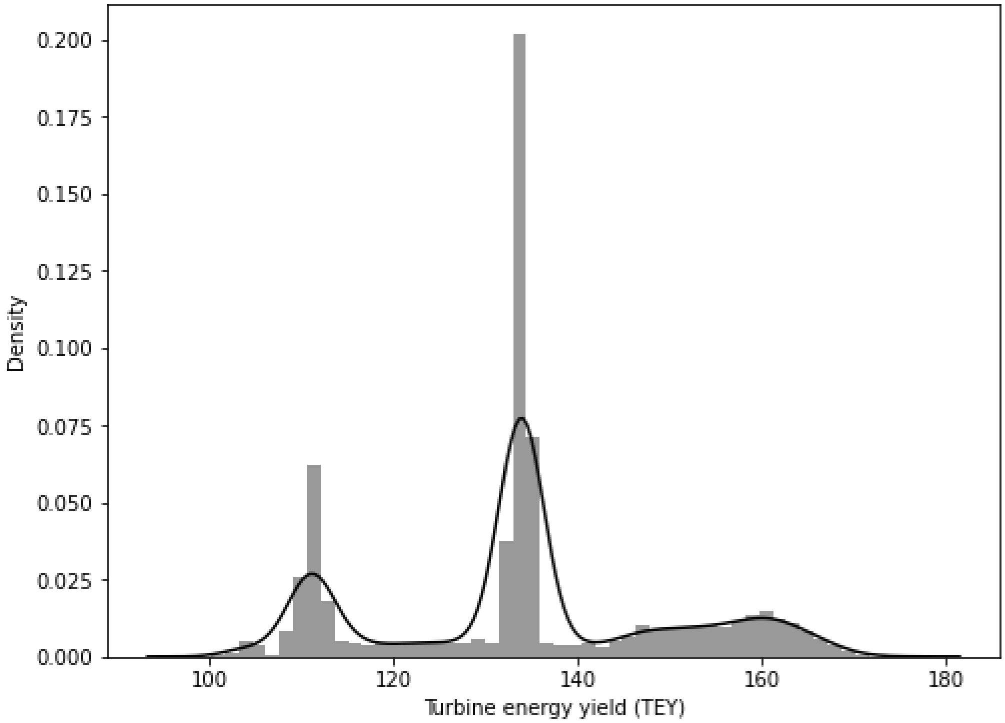
	AT	AP	AH	AFDP	GTEP	TIT	
count	15039.000000	15039.00000	15039.000000	15039.000000	15039.000000	15039.000000	15039.(
mean	17.764381	1013.19924	79.124174	4.200294	25.419061	1083.798770	545.3
std	7.574323	6.41076	13.793439	0.760197	4.173916	16.527806	7.8
min	0.522300	985.85000	30.344000	2.087400	17.878000	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	550.0
max	34.929000	1034.20000	100.200000	7.610600	37.402000	1100.800000	550.6

In [8]:
```
sns.pairplot(turbine_data)
```

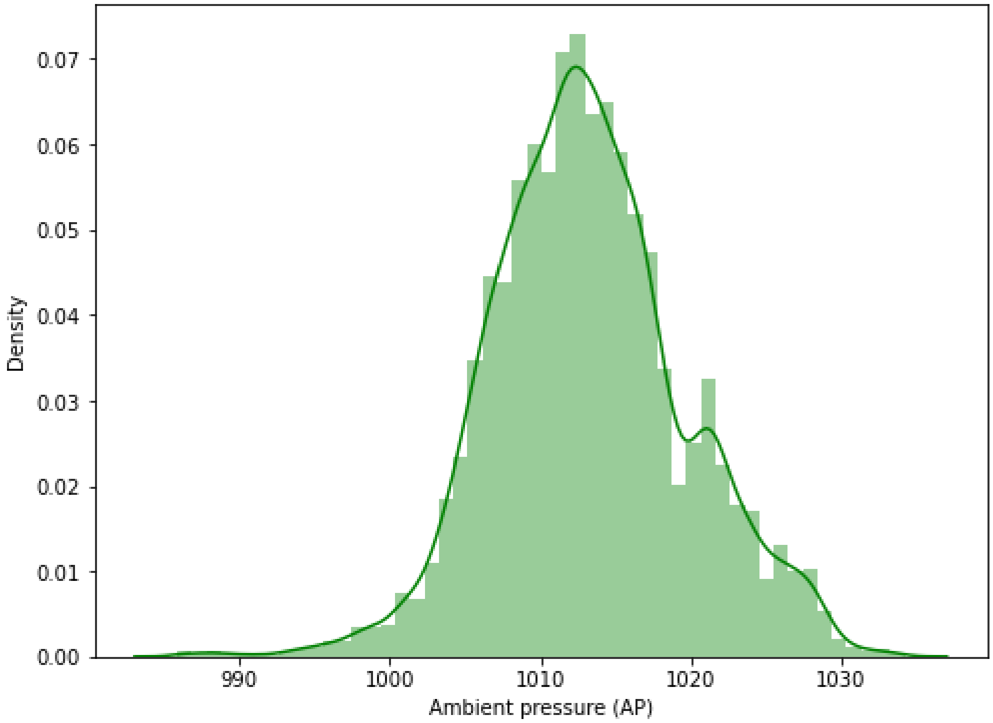
Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x1037956a250>
```



```
In [9]: plt.figure(figsize=(8,6))
        sns.distplot(x = turbine_data['TEY'],axlabel='Turbine energy yield (TEY)',color='
        plt.show()
```

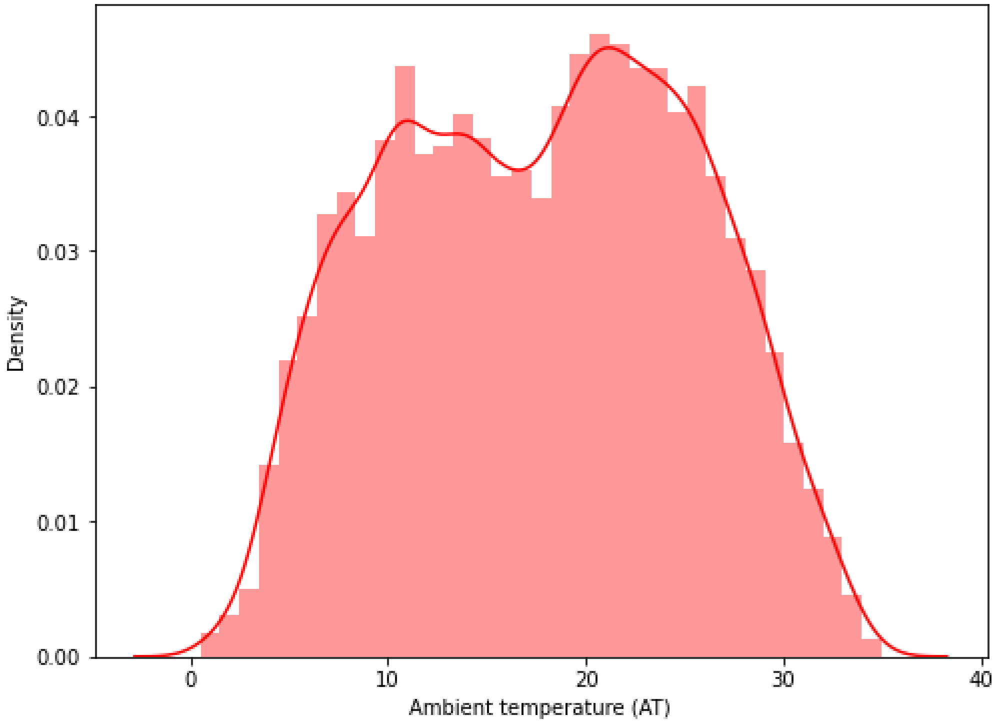


In [10]:
```
plt.figure(figsize=(8,6))
sns.distplot(x = turbine_data['AP'],axlabel='Ambient pressure (AP)',color='green'
plt.show()
```

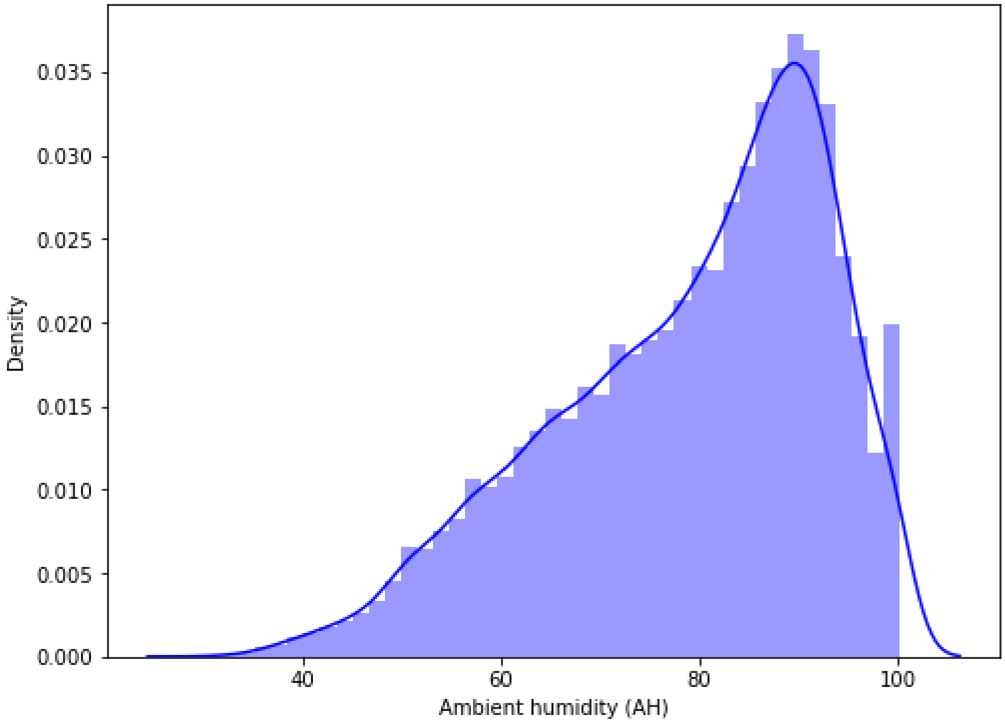


In [11]:

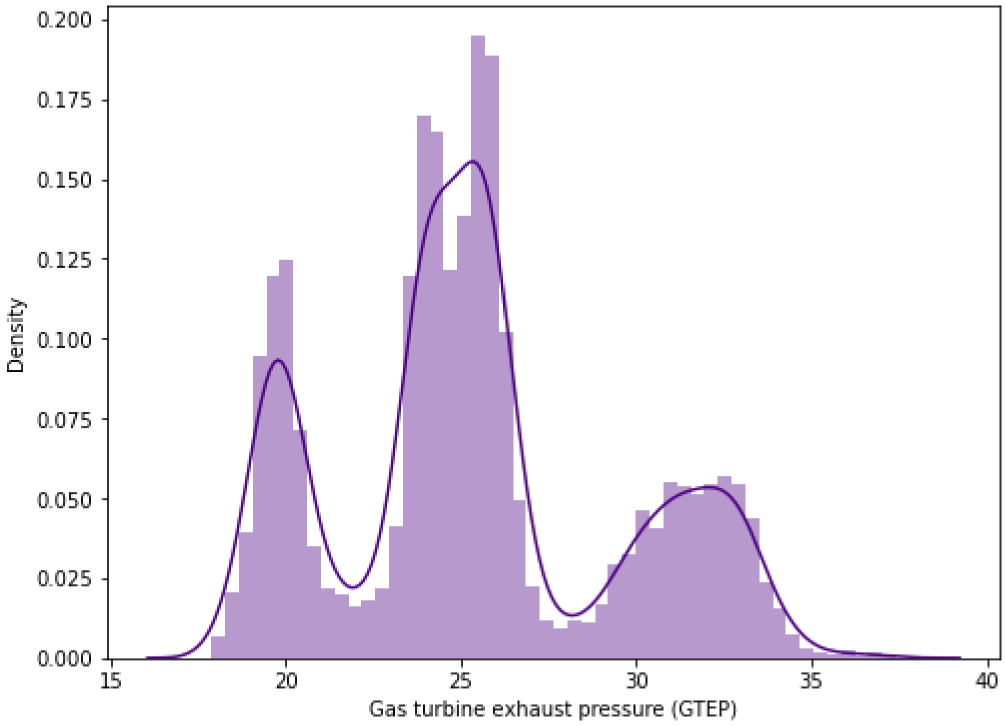
```
plt.figure(figsize=(8,6))
sns.distplot(x = turbine_data['AT'],axlabel='Ambient temperature (AT)',color='red
plt.show()
```



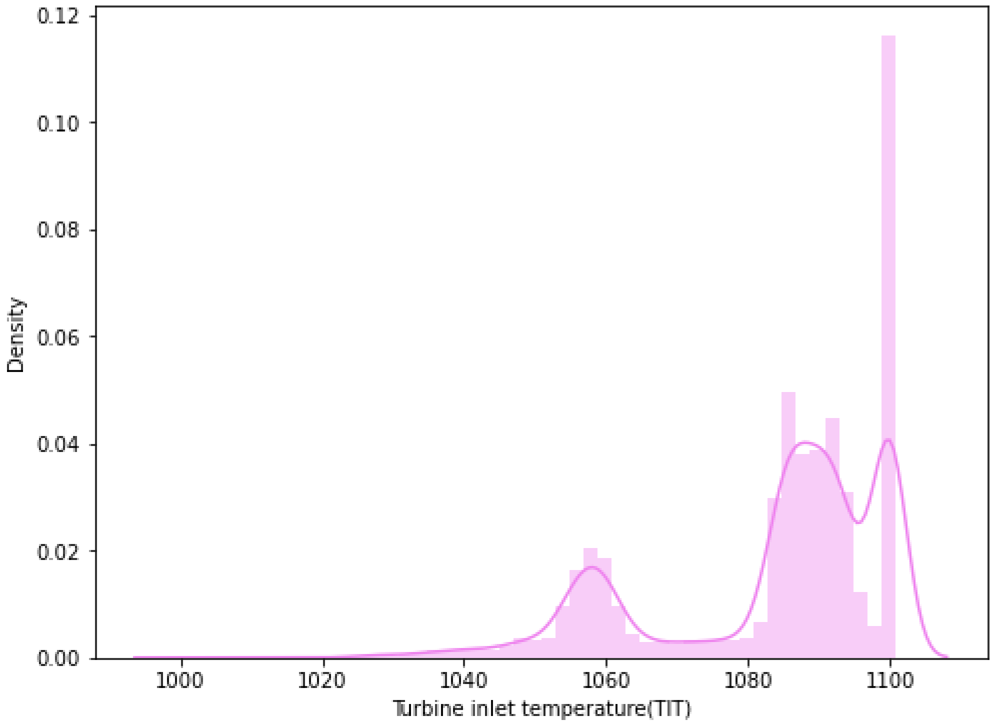
In [12]:
```
plt.figure(figsize=(8,6))
sns.distplot(x = turbine_data['AH'],axlabel='Ambient humidity (AH)',color='blue')
plt.show()
```



```
In [13]: #Gas turbine exhaust pressure (GTEP)
         plt.figure(figsize=(8,6))
         sns.distplot(x = turbine_data['GTEP'],axlabel='Gas turbine exhaust pressure (GTEP
         plt.show()
```

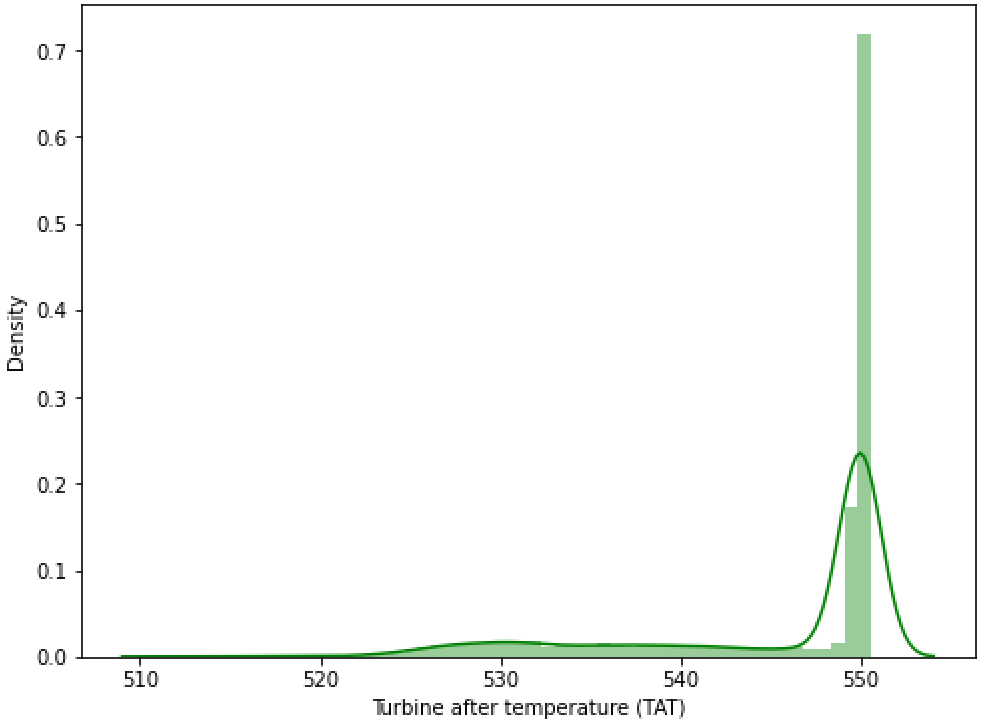


```
In [14]: #Turbine inlet temperature (TIT)
         plt.figure(figsize=(8,6))
         sns.distplot(x = turbine_data['TIT'],axlabel='Turbine inlet temperature(TIT)',col
         plt.show()
```



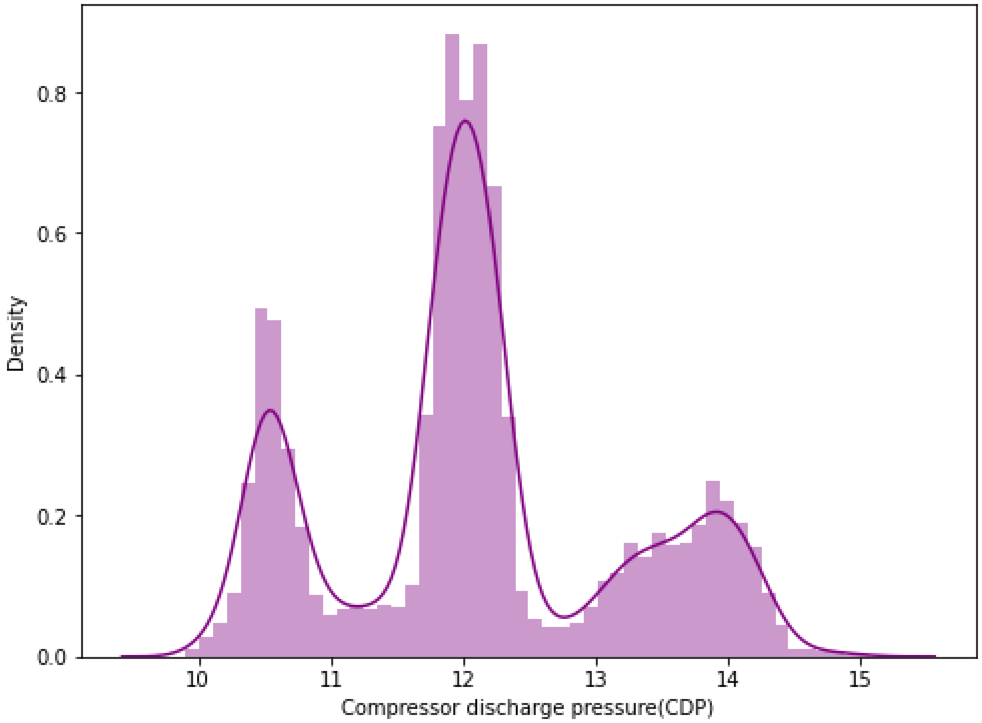
In [15]:

```
plt.figure(figsize=(8,6))
sns.distplot(x = turbine_data['TAT'],axlabel='Turbine after temperature (TAT)',co
plt.show()
```

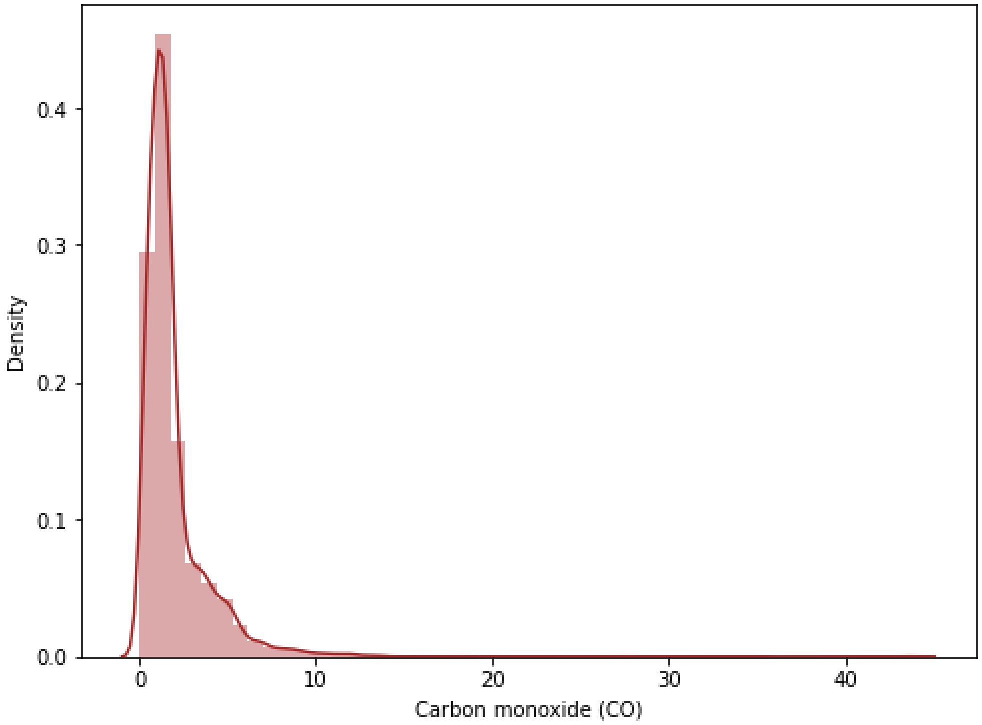


In [16]:

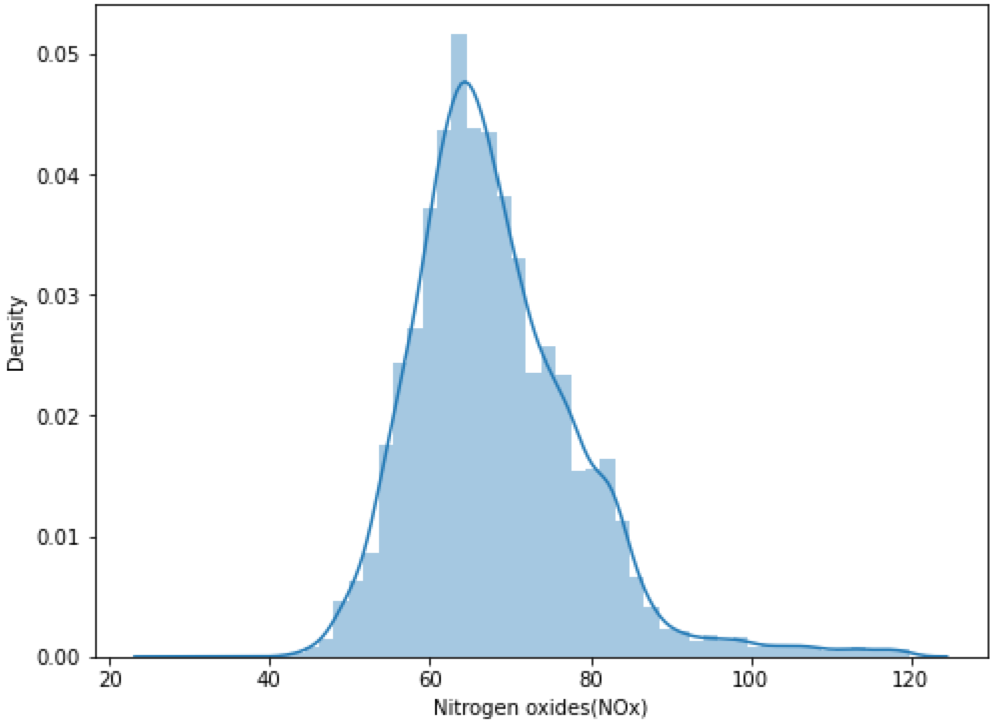
```
plt.figure(figsize=(8,6))
sns.distplot(x = turbine_data['CDP'],axlabel='Compressor discharge pressure(CDP)'
plt.show()
```



In [17]:
```
plt.figure(figsize=(8,6))
sns.distplot(x = turbine_data['CO'],axlabel='Carbon monoxide (CO)',color='brown')
plt.show()
```



In [18]:
```
plt.figure(figsize=(8,6))
sns.distplot(x = turbine_data['NOX'],axlabel='Nitrogen oxides(NOx)')
plt.show()
```



Model Building

In [19]:
```
x = turbine_data.drop(labels='TEY',axis=1)
y = turbine_data[['TEY']]
```

In [20]:

```
x
```

Out[20]:

	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 [21]:

```
x.head()
```

Out[21]:

	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

In [22]:
```
y
```

Out[22]:
```
          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 [23]:
```
y.head()
```

Out[23]:
```
      TEY
0  114.70
1  114.72
2  114.71
3  114.72
4  114.72
```

Covert input data in standard form

In [24]:
```
sclar = MinMaxScaler()
scale_data = sclar.fit_transform(x)
scale_data
```

Out[24]:
```
array([[0.18418215, 0.45604964, 0.95131413, ..., 0.1353398 , 0.07152212,
        0.59654817],
       [0.18201978, 0.4663909 , 0.95588067, ..., 0.13398756, 0.07337235,
        0.59713433],
       [0.18529531, 0.47466391, 0.93900309, ..., 0.13456709, 0.07257648,
        0.59379104],
       ...,
       [0.19596183, 0.4229576 , 0.98992213, ..., 0.11177221, 0.18055195,
        0.68544912],
       [0.18844295, 0.43329886, 0.98293633, ..., 0.12143106, 0.14169257,
        0.71057802],
       [0.18617304, 0.44157187, 0.96182146, ..., 0.1310899 , 0.11294597,
        0.70266486]])
```

In [25]:
```
x_train,x_test, y_train, y_test = train_test_split(scale_data,y,test_size=0.3,ran
```

In [26]:
```
x_train.shape,y_train.shape
```

Out[26]:
```
((10527, 10), (10527, 1))
```

In [27]:
```
x_test.shape,y_test.shape
```

Out[27]:
```
((4512, 10), (4512, 1))
```

Model Training

Turning Hyperparameter: Batch Size and epochs

In [28]:
```
def regression_model():
    model = Sequential()
    model.add(Dense(12, input_dim=10,kernel_initializer='uniform',activation='rel
    model.add(Dropout(0.2))   #The Dropout layer randomly sets input units to 0 w
    model.add(Dense(8 ,kernel_initializer='uniform',activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(1,kernel_initializer = 'uniform',activation = 'sigmoid'))

    oppti = Adam(learning_rate=0.001)
    model.compile(loss='mean_squared_error',optimizer = oppti,metrics=['mae','mse
    return model
```

In [29]:

```
model1 = KerasRegressor(build_fn=regression_model, verbose=0)
batch_size = [10,50,100]
epochs = [40,70,100]
param_grid = dict(batch_size = batch_size,epochs = epochs)
gsv = GridSearchCV(estimator=model1, param_grid=param_grid, cv = KFold(),verbose=
grid_res = gsv.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
[CV 1/5] END .....batch_size=10, epochs=40;, score=-17983.838 total time=  27.7
s
[CV 2/5] END .....batch_size=10, epochs=40;, score=-17884.646 total time=  20.9
s
[CV 3/5] END .....batch_size=10, epochs=40;, score=-17947.232 total time=  21.0
s
[CV 4/5] END .....batch_size=10, epochs=40;, score=-18139.010 total time=  22.7
s
[CV 5/5] END .....batch_size=10, epochs=40;, score=-18041.299 total time=  20.6
s
[CV 1/5] END .....batch_size=10, epochs=70;, score=-17983.838 total time=  34.6
s
[CV 2/5] END .....batch_size=10, epochs=70;, score=-17884.646 total time=  36.9
s
[CV 3/5] END .....batch_size=10, epochs=70;, score=-17947.232 total time=  34.1
s
[CV 4/5] END .....batch_size=10, epochs=70;, score=-18139.010 total time=  34.1
s
[CV 5/5] END .....batch_size=10, epochs=70;, score=-18041.299 total time=  37.3
s
[CV 1/5] END ....batch_size=10, epochs=100;, score=-17983.838 total time=  51.7
s
[CV 2/5] END ....batch_size=10, epochs=100;, score=-17884.646 total time=  50.8
s
[CV 3/5] END ....batch_size=10, epochs=100;, score=-17947.232 total time=  53.4
s
[CV 4/5] END ....batch_size=10, epochs=100;, score=-18139.010 total time=  52.6
s
[CV 5/5] END ....batch_size=10, epochs=100;, score=-18041.299 total time=  55.5
s
[CV 1/5] END .....batch_size=50, epochs=40;, score=-17983.840 total time=   4.9
s
[CV 2/5] END .....batch_size=50, epochs=40;, score=-17884.643 total time=   4.8
s
[CV 3/5] END .....batch_size=50, epochs=40;, score=-17947.230 total time=   5.8
s
[CV 4/5] END .....batch_size=50, epochs=40;, score=-18139.010 total time=   5.3
s
[CV 5/5] END .....batch_size=50, epochs=40;, score=-18041.297 total time=   5.2
s
[CV 1/5] END .....batch_size=50, epochs=70;, score=-17983.840 total time=   8.4
s
[CV 2/5] END .....batch_size=50, epochs=70;, score=-17884.643 total time=   8.0
s
[CV 3/5] END .....batch_size=50, epochs=70;, score=-17947.230 total time=   8.0
s
[CV 4/5] END .....batch_size=50, epochs=70;, score=-18139.010 total time=   9.2
s
[CV 5/5] END .....batch_size=50, epochs=70;, score=-18041.297 total time=   9.5
```

```
s
[CV 1/5] END ....batch_size=50, epochs=100;, score=-17983.840 total time=  10.9
s
[CV 2/5] END ....batch_size=50, epochs=100;, score=-17884.643 total time=  11.1
s
[CV 3/5] END ....batch_size=50, epochs=100;, score=-17947.230 total time=  11.0
s
[CV 4/5] END ....batch_size=50, epochs=100;, score=-18139.010 total time=  10.8
s
[CV 5/5] END ....batch_size=50, epochs=100;, score=-18041.297 total time=  11.3
s
[CV 1/5] END ....batch_size=100, epochs=40;, score=-17983.834 total time=   2.6
s
[CV 2/5] END ....batch_size=100, epochs=40;, score=-17884.643 total time=   3.0
s
[CV 3/5] END ....batch_size=100, epochs=40;, score=-17947.230 total time=   2.8
s
[CV 4/5] END ....batch_size=100, epochs=40;, score=-18139.010 total time=   3.0
s
[CV 5/5] END ....batch_size=100, epochs=40;, score=-18041.297 total time=   3.6
s
[CV 1/5] END ....batch_size=100, epochs=70;, score=-17983.834 total time=   5.2
s
[CV 2/5] END ....batch_size=100, epochs=70;, score=-17884.643 total time=   4.2
s
[CV 3/5] END ....batch_size=100, epochs=70;, score=-17947.230 total time=   4.6
s
[CV 4/5] END ....batch_size=100, epochs=70;, score=-18139.010 total time=   4.3
s
[CV 5/5] END ....batch_size=100, epochs=70;, score=-18041.297 total time=   4.2
s
[CV 1/5] END ...batch_size=100, epochs=100;, score=-17983.834 total time=   5.7
s
[CV 2/5] END ...batch_size=100, epochs=100;, score=-17884.643 total time=   7.3
s
[CV 3/5] END ...batch_size=100, epochs=100;, score=-17947.230 total time=   6.7
s
[CV 4/5] END ...batch_size=100, epochs=100;, score=-18139.010 total time=   8.1
s
[CV 5/5] END ...batch_size=100, epochs=100;, score=-18041.297 total time=   6.5
s
```

In [30]:

```
print(grid_res.best_score_,grid_res.best_params_)
```

```
-17999.202734375 {'batch_size': 100, 'epochs': 40}
```

Tuning of Hyperparameter :Activation Function & Kernel Initializer

In [31]:

```
def regression_model(activation_function,inti):
    model = Sequential()
    model.add(Dense(12, input_dim=10,kernel_initializer='uniform',activation='rel
    model.add(Dropout(0.0))
    model.add(Dense(8 ,kernel_initializer='uniform',activation='relu'))
    model.add(Dropout(0.0))
    model.add(Dense(1,kernel_initializer = 'uniform',activation = 'sigmoid'))
    adam = Adam(learning_rate=0.1)
    model.compile(loss='mean_squared_error',optimizer=adam, metrics=['mae','mse']
    return model
```

In [32]:

```
model = KerasRegressor(build_fn=regression_model,batch_size = 100,epochs = 40,ver
activation_function = ['relu', 'tanh','softmax','linear']
inti = ['zero', 'uniform','normal']
param_grid = dict(activation_function = activation_function,inti = inti)
gsv_m = GridSearchCV(estimator=model,param_grid=param_grid,cv=KFold(),verbose = 5
grid_resu = gsv_m.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
[CV 1/5] END activation_function=relu, inti=zero;, score=-17983.834 total time=
2.6s
[CV 2/5] END activation_function=relu, inti=zero;, score=-17884.643 total time=
2.4s
[CV 3/5] END activation_function=relu, inti=zero;, score=-17947.230 total time=
2.6s
[CV 4/5] END activation_function=relu, inti=zero;, score=-18139.010 total time=
2.9s
[CV 5/5] END activation_function=relu, inti=zero;, score=-18041.297 total time=
3.1s
[CV 1/5] END activation_function=relu, inti=uniform;, score=-17983.834 total ti
me=   2.6s
[CV 2/5] END activation_function=relu, inti=uniform;, score=-17884.643 total ti
me=   2.5s
[CV 3/5] END activation_function=relu, inti=uniform;, score=-17947.230 total ti
me=   2.7s
[CV 4/5] END activation_function=relu, inti=uniform;, score=-18139.010 total ti
me=   2.7s
[CV 5/5] END activation_function=relu, inti=uniform;, score=-18041.297 total ti
me=   2.6s
[CV 1/5] END activation_function=relu, inti=normal;, score=-17983.834 total tim
e=   2.5s
[CV 2/5] END activation_function=relu, inti=normal;, score=-17884.643 total tim
e=   2.7s
[CV 3/5] END activation_function=relu, inti=normal;, score=-17947.230 total tim
e=   2.5s
[CV 4/5] END activation_function=relu, inti=normal;, score=-18139.010 total tim
e=   2.4s
[CV 5/5] END activation_function=relu, inti=normal;, score=-18041.297 total tim
e=   2.7s
[CV 1/5] END activation_function=tanh, inti=zero;, score=-17983.834 total time=
2.6s
[CV 2/5] END activation_function=tanh, inti=zero;, score=-17884.643 total time=
2.6s
[CV 3/5] END activation_function=tanh, inti=zero;, score=-17947.230 total time=
2.5s
[CV 4/5] END activation_function=tanh, inti=zero;, score=-18139.010 total time=
2.8s
[CV 5/5] END activation_function=tanh, inti=zero;, score=-18041.297 total time=
2.6s
[CV 1/5] END activation_function=tanh, inti=uniform;, score=-17983.834 total ti
me=   2.6s
[CV 2/5] END activation_function=tanh, inti=uniform;, score=-17884.643 total ti
me=   2.7s
[CV 3/5] END activation_function=tanh, inti=uniform;, score=-17947.230 total ti
me=   2.9s
[CV 4/5] END activation_function=tanh, inti=uniform;, score=-18139.010 total ti
me=   2.4s
[CV 5/5] END activation_function=tanh, inti=uniform;, score=-18041.297 total ti
```

```
me=   2.7s
[CV 1/5] END activation_function=tanh, inti=normal;, score=-17983.834 total tim
e=   3.1s
[CV 2/5] END activation_function=tanh, inti=normal;, score=-17884.643 total tim
e=   2.6s
[CV 3/5] END activation_function=tanh, inti=normal;, score=-17947.230 total tim
e=   2.6s
[CV 4/5] END activation_function=tanh, inti=normal;, score=-18139.010 total tim
e=   2.7s
[CV 5/5] END activation_function=tanh, inti=normal;, score=-18041.297 total tim
e=   2.5s
[CV 1/5] END activation_function=softmax, inti=zero;, score=-17983.834 total ti
me=   2.7s
[CV 2/5] END activation_function=softmax, inti=zero;, score=-17884.643 total ti
me=   2.6s
[CV 3/5] END activation_function=softmax, inti=zero;, score=-17947.230 total ti
me=   2.6s
[CV 4/5] END activation_function=softmax, inti=zero;, score=-18139.010 total ti
me=   2.8s
[CV 5/5] END activation_function=softmax, inti=zero;, score=-18041.297 total ti
me=   2.6s
[CV 1/5] END activation_function=softmax, inti=uniform;, score=-17983.834 total
time=   2.6s
[CV 2/5] END activation_function=softmax, inti=uniform;, score=-17884.643 total
time=   2.6s
[CV 3/5] END activation_function=softmax, inti=uniform;, score=-17947.230 total
time=   2.7s
[CV 4/5] END activation_function=softmax, inti=uniform;, score=-18139.010 total
time=   2.4s
[CV 5/5] END activation_function=softmax, inti=uniform;, score=-18041.305 total
time=   2.5s
[CV 1/5] END activation_function=softmax, inti=normal;, score=-17983.834 total
time=   2.7s
[CV 2/5] END activation_function=softmax, inti=normal;, score=-17884.643 total
time=   2.6s
[CV 3/5] END activation_function=softmax, inti=normal;, score=-17947.230 total
time=   2.9s
[CV 4/5] END activation_function=softmax, inti=normal;, score=-18139.010 total
time=   2.8s
[CV 5/5] END activation_function=softmax, inti=normal;, score=-18041.297 total
time=   2.4s
[CV 1/5] END activation_function=linear, inti=zero;, score=-17983.834 total tim
e=   2.6s
[CV 2/5] END activation_function=linear, inti=zero;, score=-17884.643 total tim
e=   3.0s
[CV 3/5] END activation_function=linear, inti=zero;, score=-17947.230 total tim
e=   2.6s
[CV 4/5] END activation_function=linear, inti=zero;, score=-18139.010 total tim
e=   2.8s
[CV 5/5] END activation_function=linear, inti=zero;, score=-18041.297 total tim
e=   2.8s
[CV 1/5] END activation_function=linear, inti=uniform;, score=-17983.834 total
time=   2.7s
[CV 2/5] END activation_function=linear, inti=uniform;, score=-17884.643 total
time=   2.5s
[CV 3/5] END activation_function=linear, inti=uniform;, score=-17947.230 total
time=   2.6s
```

```
[CV 4/5] END activation_function=linear, inti=uniform;, score=-18139.010 total
time=   2.4s
[CV 5/5] END activation_function=linear, inti=uniform;, score=-18041.297 total
time=   2.5s
[CV 1/5] END activation_function=linear, inti=normal;, score=-17983.834 total t
ime=   2.8s
[CV 2/5] END activation_function=linear, inti=normal;, score=-17884.643 total t
ime=   2.7s
[CV 3/5] END activation_function=linear, inti=normal;, score=-17947.230 total t
ime=   3.1s
[CV 4/5] END activation_function=linear, inti=normal;, score=-18139.010 total t
ime=   2.6s
[CV 5/5] END activation_function=linear, inti=normal;, score=-18041.297 total t
ime=   2.7s
```

In [33]:

```
print(grid_resu.best_score_,grid_resu.best_params_)
```

```
-17999.202734375 {'activation_function': 'relu', 'inti': 'zero'}
```

Tuning of Hyperparameter :Activation Function & Kernel Initializer

In [34]:

```
def regression_model(activation_function,inti):
    model = Sequential()
    model.add(Dense(12, input_dim=10,kernel_initializer='uniform',activation='rel
    model.add(Dropout(0.0))
    model.add(Dense(8 ,kernel_initializer='uniform',activation='relu'))
    model.add(Dropout(0.0))
    model.add(Dense(1,kernel_initializer = 'uniform',activation = 'sigmoid'))
    adam = Adam(learning_rate=0.1)
    model.compile(loss='mean_squared_error',optimizer=adam, metrics=['mae','mse']
    return model
```

In [35]:
```
model = KerasRegressor(build_fn=regression_model,batch_size = 100,epochs = 40,ver
activation_function = ['relu', 'tanh','softmax','linear']
inti = ['zero', 'uniform','normal']
param_grid = dict(activation_function = activation_function,inti = inti)
gsv_m = GridSearchCV(estimator=model,param_grid=param_grid,cv=KFold(),verbose = 5
grid_resu = gsv_m.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
[CV 1/5] END activation_function=relu, inti=zero;, score=-17983.834 total time=
2.6s
[CV 2/5] END activation_function=relu, inti=zero;, score=-17884.643 total time=
2.9s
[CV 3/5] END activation_function=relu, inti=zero;, score=-17947.230 total time=
3.0s
[CV 4/5] END activation_function=relu, inti=zero;, score=-18139.010 total time=
2.8s
[CV 5/5] END activation_function=relu, inti=zero;, score=-18041.297 total time=
2.5s
[CV 1/5] END activation_function=relu, inti=uniform;, score=-17983.834 total ti
me=   2.7s
[CV 2/5] END activation_function=relu, inti=uniform;, score=-17884.643 total ti
me=   3.5s
[CV 3/5] END activation_function=relu, inti=uniform;, score=-17947.230 total ti
me=   2.5s
[CV 4/5] END activation_function=relu, inti=uniform;, score=-18139.010 total ti
me=   2.5s
[CV 5/5] END activation_function=relu, inti=uniform;, score=-18041.297 total ti
me=   2.5s
[CV 1/5] END activation_function=relu, inti=normal;, score=-17983.834 total tim
e=   2.6s
[CV 2/5] END activation_function=relu, inti=normal;, score=-17884.643 total tim
e=   2.5s
[CV 3/5] END activation_function=relu, inti=normal;, score=-17947.230 total tim
e=   2.5s
[CV 4/5] END activation_function=relu, inti=normal;, score=-18139.010 total tim
e=   2.6s
[CV 5/5] END activation_function=relu, inti=normal;, score=-18041.297 total tim
e=   2.4s
[CV 1/5] END activation_function=tanh, inti=zero;, score=-17983.834 total time=
2.6s
[CV 2/5] END activation_function=tanh, inti=zero;, score=-17884.643 total time=
2.3s
[CV 3/5] END activation_function=tanh, inti=zero;, score=-17947.230 total time=
2.4s
[CV 4/5] END activation_function=tanh, inti=zero;, score=-18139.010 total time=
2.7s
[CV 5/5] END activation_function=tanh, inti=zero;, score=-18041.297 total time=
2.5s
[CV 1/5] END activation_function=tanh, inti=uniform;, score=-17983.834 total ti
me=   2.6s
[CV 2/5] END activation_function=tanh, inti=uniform;, score=-17884.643 total ti
me=   2.6s
[CV 3/5] END activation_function=tanh, inti=uniform;, score=-17947.230 total ti
me=   2.7s
[CV 4/5] END activation_function=tanh, inti=uniform;, score=-18139.010 total ti
me=   2.6s
[CV 5/5] END activation_function=tanh, inti=uniform;, score=-18041.297 total ti
```

```
me=   2.6s
[CV 1/5] END activation_function=tanh, inti=normal;, score=-17983.834 total tim
e=   2.7s
[CV 2/5] END activation_function=tanh, inti=normal;, score=-17884.643 total tim
e=   2.5s
[CV 3/5] END activation_function=tanh, inti=normal;, score=-17947.230 total tim
e=   3.1s
[CV 4/5] END activation_function=tanh, inti=normal;, score=-18139.010 total tim
e=   2.6s
[CV 5/5] END activation_function=tanh, inti=normal;, score=-18041.297 total tim
e=   2.6s
[CV 1/5] END activation_function=softmax, inti=zero;, score=-17983.834 total ti
me=   2.6s
[CV 2/5] END activation_function=softmax, inti=zero;, score=-17884.643 total ti
me=   3.5s
[CV 3/5] END activation_function=softmax, inti=zero;, score=-17947.230 total ti
me=   3.5s
[CV 4/5] END activation_function=softmax, inti=zero;, score=-18139.010 total ti
me=   2.7s
[CV 5/5] END activation_function=softmax, inti=zero;, score=-18041.297 total ti
me=   2.9s
[CV 1/5] END activation_function=softmax, inti=uniform;, score=-17983.834 total
time=   2.7s
[CV 2/5] END activation_function=softmax, inti=uniform;, score=-17884.643 total
time=   2.8s
[CV 3/5] END activation_function=softmax, inti=uniform;, score=-17947.230 total
time=   2.7s
[CV 4/5] END activation_function=softmax, inti=uniform;, score=-18139.010 total
time=   2.5s
[CV 5/5] END activation_function=softmax, inti=uniform;, score=-18041.297 total
time=   2.5s
[CV 1/5] END activation_function=softmax, inti=normal;, score=-17983.834 total
time=   2.9s
[CV 2/5] END activation_function=softmax, inti=normal;, score=-17884.643 total
time=   2.6s
[CV 3/5] END activation_function=softmax, inti=normal;, score=-17947.230 total
time=   2.6s
[CV 4/5] END activation_function=softmax, inti=normal;, score=-18139.010 total
time=   2.6s
[CV 5/5] END activation_function=softmax, inti=normal;, score=-18041.297 total
time=   2.6s
[CV 1/5] END activation_function=linear, inti=zero;, score=-17983.834 total tim
e=   2.6s
[CV 2/5] END activation_function=linear, inti=zero;, score=-17884.643 total tim
e=   2.6s
[CV 3/5] END activation_function=linear, inti=zero;, score=-17947.230 total tim
e=   2.8s
[CV 4/5] END activation_function=linear, inti=zero;, score=-18139.010 total tim
e=   2.8s
[CV 5/5] END activation_function=linear, inti=zero;, score=-18041.297 total tim
e=   3.0s
[CV 1/5] END activation_function=linear, inti=uniform;, score=-17983.834 total
time=   2.4s
[CV 2/5] END activation_function=linear, inti=uniform;, score=-17884.643 total
time=   2.5s
[CV 3/5] END activation_function=linear, inti=uniform;, score=-17947.230 total
time=   2.9s
```

```
[CV 4/5] END activation_function=linear, inti=uniform;, score=-18139.010 total
time=   3.0s
[CV 5/5] END activation_function=linear, inti=uniform;, score=-18041.297 total
time=   2.4s
[CV 1/5] END activation_function=linear, inti=normal;, score=-17983.834 total t
ime=   2.4s
[CV 2/5] END activation_function=linear, inti=normal;, score=-17884.643 total t
ime=   2.5s
[CV 3/5] END activation_function=linear, inti=normal;, score=-17947.230 total t
ime=   2.5s
[CV 4/5] END activation_function=linear, inti=normal;, score=-18139.010 total t
ime=   2.7s
[CV 5/5] END activation_function=linear, inti=normal;, score=-18041.297 total t
ime=   2.5s
```

In [36]:
```
print(grid_resu.best_score_,grid_resu.best_params_)
```

```
-17999.202734375 {'activation_function': 'relu', 'inti': 'zero'}
```

Tuning of Hyperparameter :Number of Neurons in hidden layer

In [37]:
```
def regression_model(neuron1,neuron2):
    model = Sequential()
    model.add(Dense(12,input_dim = 10,kernel_initializer='uniform',activation='re
    model.add(Dropout(0.0))
    model.add(Dense(8,kernel_initializer = 'uniform',activation = 'relu'))
    model.add(Dropout(0.0))
    model.add(Dense(1,kernel_initializer='uniform',activation= 'sigmoid'))
    adam = Adam(learning_rate= 0.1)
    model.compile(loss='mean_squared_error',optimizer=adam,metrics=['mae','mse'])
    return model
```

In [38]:
```
model = KerasRegressor(build_fn=regression_model,batch_size = 100,epochs = 40,ver
neuron1 = [16,12,8]
neuron2 = [12,8,4]
param_grid = dict(neuron1 = neuron1,neuron2 = neuron2)
gsvp = GridSearchCV(estimator = model,param_grid = param_grid,cv=KFold(),verbose=
grid_result = gsvp.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
[CV 1/5] END .......neuron1=16, neuron2=12;, score=-17983.834 total time=   2.5
s
[CV 2/5] END .......neuron1=16, neuron2=12;, score=-17884.643 total time=   2.5
s
[CV 3/5] END .......neuron1=16, neuron2=12;, score=-17947.230 total time=   2.7
s
[CV 4/5] END .......neuron1=16, neuron2=12;, score=-18139.010 total time=   2.6
s
[CV 5/5] END .......neuron1=16, neuron2=12;, score=-18041.297 total time=   2.9
s
[CV 1/5] END ........neuron1=16, neuron2=8;, score=-17983.834 total time=   2.9
s
[CV 2/5] END ........neuron1=16, neuron2=8;, score=-17884.643 total time=   2.9
s
[CV 3/5] END ........neuron1=16, neuron2=8;, score=-17947.230 total time=   2.8
s
[CV 4/5] END ........neuron1=16, neuron2=8;, score=-18139.010 total time=   2.9
s
[CV 5/5] END ........neuron1=16, neuron2=8;, score=-18041.297 total time=   3.2
s
[CV 1/5] END ........neuron1=16, neuron2=4;, score=-17983.834 total time=   2.6
s
[CV 2/5] END ........neuron1=16, neuron2=4;, score=-17884.643 total time=   2.6
s
[CV 3/5] END ........neuron1=16, neuron2=4;, score=-17947.230 total time=   2.7
s
[CV 4/5] END ........neuron1=16, neuron2=4;, score=-18139.010 total time=   2.5
s
[CV 5/5] END ........neuron1=16, neuron2=4;, score=-18041.297 total time=   2.4
s
[CV 1/5] END .......neuron1=12, neuron2=12;, score=-17983.834 total time=   2.5
s
[CV 2/5] END .......neuron1=12, neuron2=12;, score=-17884.643 total time=   2.5
s
[CV 3/5] END .......neuron1=12, neuron2=12;, score=-17947.230 total time=   2.7
s
[CV 4/5] END .......neuron1=12, neuron2=12;, score=-18139.010 total time=   2.7
s
[CV 5/5] END .......neuron1=12, neuron2=12;, score=-18041.297 total time=   2.4
s
[CV 1/5] END ........neuron1=12, neuron2=8;, score=-17983.834 total time=   2.4
s
[CV 2/5] END ........neuron1=12, neuron2=8;, score=-17884.643 total time=   2.7
s
[CV 3/5] END ........neuron1=12, neuron2=8;, score=-17947.246 total time=   2.7
s
[CV 4/5] END ........neuron1=12, neuron2=8;, score=-18139.010 total time=   2.9
s
[CV 5/5] END ........neuron1=12, neuron2=8;, score=-18041.297 total time=   2.7
```

```
s
[CV 1/5] END ........neuron1=12, neuron2=4;, score=-17983.834 total time=   2.9
s
[CV 2/5] END ........neuron1=12, neuron2=4;, score=-17884.643 total time=   2.5
s
[CV 3/5] END ........neuron1=12, neuron2=4;, score=-17947.230 total time=   2.6
s
[CV 4/5] END ........neuron1=12, neuron2=4;, score=-18139.010 total time=   2.6
s
[CV 5/5] END ........neuron1=12, neuron2=4;, score=-18041.297 total time=   2.6
s
[CV 1/5] END ........neuron1=8, neuron2=12;, score=-17983.834 total time=   2.8
s
[CV 2/5] END ........neuron1=8, neuron2=12;, score=-17884.643 total time=   3.2
s
[CV 3/5] END ........neuron1=8, neuron2=12;, score=-17947.230 total time=   2.8
s
[CV 4/5] END ........neuron1=8, neuron2=12;, score=-18139.010 total time=   2.6
s
[CV 5/5] END ........neuron1=8, neuron2=12;, score=-18041.297 total time=   2.6
s
[CV 1/5] END .........neuron1=8, neuron2=8;, score=-17983.834 total time=   2.7
s
[CV 2/5] END .........neuron1=8, neuron2=8;, score=-17884.643 total time=   2.8
s
[CV 3/5] END .........neuron1=8, neuron2=8;, score=-17947.230 total time=   2.6
s
[CV 4/5] END .........neuron1=8, neuron2=8;, score=-18139.010 total time=   2.9
s
[CV 5/5] END .........neuron1=8, neuron2=8;, score=-18041.297 total time=   2.9
s
[CV 1/5] END .........neuron1=8, neuron2=4;, score=-17983.834 total time=   2.6
s
[CV 2/5] END .........neuron1=8, neuron2=4;, score=-17884.643 total time=   2.4
s
[CV 3/5] END .........neuron1=8, neuron2=4;, score=-17947.248 total time=   2.6
s
[CV 4/5] END .........neuron1=8, neuron2=4;, score=-18139.010 total time=   2.6
s
[CV 5/5] END .........neuron1=8, neuron2=4;, score=-18041.297 total time=   2.6
s
```

```
In [39]: print(grid_result.best_score_,grid_result.best_params_)
```

```
-17999.202734375 {'neuron1': 16, 'neuron2': 12}
```

Train a model with optimum values of hyperparameter

In [40]:

```
# Best parametes
# batch_size: 100
# epochs: 40
# dropout_rate: 0.0
# learning_rate: 0.1
# activation_function: relu
# inti: uniform
# neuron1: 16
# neuron2: 12
```

In [41]:

```
model = Sequential()
model.add(Dense(16,input_dim = 10,kernel_initializer='uniform',activation='relu')
model.add(Dropout(0.0))
model.add(Dense(12,kernel_initializer = 'uniform',activation = 'relu'))
model.add(Dropout(0.0))
model.add(Dense(1,kernel_initializer='uniform',activation='relu'))
optimizer = RMSprop(learning_rate=0.1)
model.compile(loss='mse',optimizer = optimizer,metrics=['mae','mse'])
```

In [42]:

```
model.fit(x_train,y_train,batch_size=100,epochs=40)
y_pred = model.predict(x_train)
```

```
Epoch 1/40
106/106 [==============================] - 1s 2ms/step - loss: 957.1918 - ma
e: 25.1612 - mse: 957.1918
Epoch 2/40
106/106 [==============================] - 0s 933us/step - loss: 471.0055 - m
ae: 20.8852 - mse: 471.0055
Epoch 3/40
106/106 [==============================] - 0s 1ms/step - loss: 459.4968 - ma
e: 20.7106 - mse: 459.4968
Epoch 4/40
106/106 [==============================] - 0s 1ms/step - loss: 443.7420 - ma
e: 20.3619 - mse: 443.7420
Epoch 5/40
106/106 [==============================] - 0s 914us/step - loss: 421.0845 - m
ae: 19.9112 - mse: 421.08450s - loss: 429.7915 - mae: 20.0952 - mse: 429.79
Epoch 6/40
106/106 [==============================] - 0s 855us/step - loss: 372.5999 - m
ae: 18.7258 - mse: 372.5999
Epoch 7/40
106/106 [==============================] - 0s 874us/step - loss: 287.0925 - m
ae: 16.3668 - mse: 287.0925
Epoch 8/40
106/106 [==============================] - 0s 845us/step - loss: 208.0458 - m
ae: 13.8744 - mse: 208.0458
Epoch 9/40
106/106 [==============================] - 0s 1ms/step - loss: 190.4399 - ma
e: 13.2876 - mse: 190.4399
Epoch 10/40
106/106 [==============================] - 0s 979us/step - loss: 164.6834 - m
ae: 12.2756 - mse: 164.6834
Epoch 11/40
106/106 [==============================] - 0s 940us/step - loss: 127.2675 - m
ae: 10.7806 - mse: 127.2675
Epoch 12/40
106/106 [==============================] - 0s 788us/step - loss: 96.7662 - ma
e: 9.2609 - mse: 96.7662
Epoch 13/40
106/106 [==============================] - 0s 789us/step - loss: 72.4239 - ma
e: 7.9243 - mse: 72.4239
Epoch 14/40
106/106 [==============================] - 0s 817us/step - loss: 68.0525 - ma
e: 7.7737 - mse: 68.0525
Epoch 15/40
106/106 [==============================] - 0s 883us/step - loss: 65.2592 - ma
e: 7.6092 - mse: 65.2592
Epoch 16/40
106/106 [==============================] - 0s 864us/step - loss: 63.2786 - ma
e: 7.5604 - mse: 63.2786
Epoch 17/40
106/106 [==============================] - 0s 798us/step - loss: 59.0180 - ma
e: 7.3002 - mse: 59.0180
Epoch 18/40
106/106 [==============================] - 0s 855us/step - loss: 57.6304 - ma
```

```
e: 7.2513 - mse: 57.6304
Epoch 19/40
106/106 [==============================] - 0s 988us/step - loss: 55.6810 - ma
e: 7.1736 - mse: 55.6810
Epoch 20/40
106/106 [==============================] - 0s 959us/step - loss: 52.9645 - ma
e: 7.0118 - mse: 52.9645
Epoch 21/40
106/106 [==============================] - 0s 931us/step - loss: 50.5853 - ma
e: 6.8818 - mse: 50.5853
Epoch 22/40
106/106 [==============================] - 0s 855us/step - loss: 48.8737 - ma
e: 6.7496 - mse: 48.8737
Epoch 23/40
106/106 [==============================] - 0s 807us/step - loss: 46.4756 - ma
e: 6.6156 - mse: 46.4756
Epoch 24/40
106/106 [==============================] - 0s 817us/step - loss: 44.7438 - ma
e: 6.4845 - mse: 44.7438
Epoch 25/40
106/106 [==============================] - 0s 793us/step - loss: 41.5717 - ma
e: 6.2560 - mse: 41.5717
Epoch 26/40
106/106 [==============================] - 0s 959us/step - loss: 37.8643 - ma
e: 5.9896 - mse: 37.8643
Epoch 27/40
106/106 [==============================] - 0s 950us/step - loss: 34.9719 - ma
e: 5.7120 - mse: 34.9719
Epoch 28/40
106/106 [==============================] - 0s 969us/step - loss: 34.7543 - ma
e: 5.6764 - mse: 34.7543
Epoch 29/40
106/106 [==============================] - 0s 874us/step - loss: 33.6819 - ma
e: 5.5446 - mse: 33.6819
Epoch 30/40
106/106 [==============================] - 0s 807us/step - loss: 32.4674 - ma
e: 5.4923 - mse: 32.4674
Epoch 31/40
106/106 [==============================] - 0s 779us/step - loss: 31.9515 - ma
e: 5.4293 - mse: 31.9515
Epoch 32/40
106/106 [==============================] - 0s 817us/step - loss: 30.8657 - ma
e: 5.3528 - mse: 30.8657
Epoch 33/40
106/106 [==============================] - 0s 845us/step - loss: 30.8869 - ma
e: 5.3430 - mse: 30.8869
Epoch 34/40
106/106 [==============================] - 0s 826us/step - loss: 29.5396 - ma
e: 5.2163 - mse: 29.5396
Epoch 35/40
106/106 [==============================] - 0s 921us/step - loss: 28.7110 - ma
e: 5.0833 - mse: 28.7110
Epoch 36/40
106/106 [==============================] - 0s 959us/step - loss: 27.8923 - ma
e: 5.0136 - mse: 27.8923
Epoch 37/40
106/106 [==============================] - 0s 893us/step - loss: 27.6323 - ma
```

```
e: 4.9880 - mse: 27.6323
Epoch 38/40
106/106 [==============================] - 0s 909us/step - loss: 26.0808 - ma
e: 4.8387 - mse: 26.0808
Epoch 39/40
106/106 [==============================] - 0s 788us/step - loss: 26.0210 - ma
e: 4.8556 - mse: 26.0210
Epoch 40/40
106/106 [==============================] - 0s 769us/step - loss: 25.1463 - ma
e: 4.7286 - mse: 25.1463
```

In [43]:

```
model.summary()
```

```
Model: "sequential_214"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense_642 (Dense)           (None, 16)                176       
                                                                 
 dropout_428 (Dropout)       (None, 16)                0         
                                                                 
 dense_643 (Dense)           (None, 12)                204       
                                                                 
 dropout_429 (Dropout)       (None, 12)                0         
                                                                 
 dense_644 (Dense)           (None, 1)                 13        
                                                                 
=================================================================
Total params: 393
Trainable params: 393
Non-trainable params: 0
_________________________________________________________________
```

In [44]:

```
mean_absolute_error(y_train,y_pred)
```

Out[44]:

```
2.8760812284002686
```

In [45]:

```
mean_squared_error(y_train,y_pred)
```

Out[45]:

```
9.786654752875236
```

In [46]:

```
# testing data
test_score = model.evaluate(x_test,y_test)
```

```
141/141 [==============================] - 0s 769us/step - loss: 9.8016 - mae:
2.8693 - mse: 9.8016
```

In [47]:

```
y_test_pred = model.predict(x_test)
```

In [48]:

```
mean_absolute_error(y_test,y_test_pred)
```

Out[48]:

```
2.869305136863221
```

In [49]:

```
mean_squared_error(y_test,y_test_pred)
```

Out[49]:

```
9.80161210780495
```

In [ ]:

```
```