Problem statement: predicting turbine energy yield (TEY) using ambient variables as features.

In [1]:

```
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]:

```
seed = 7
np.random.seed(seed)
gas_turbine = pd.read_csv("gas_turbines.csv")
gas_turbine
```

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]:

```
gas_turbine.shape
```

Out[3]:

```
(15039, 11)
```

In [4]:

```
gas_turbine.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]:

```
gas_turbine.dtypes
```

Out[5]:

```
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 [6]:

```
gas_turbine.describe()
```

Out[6]:

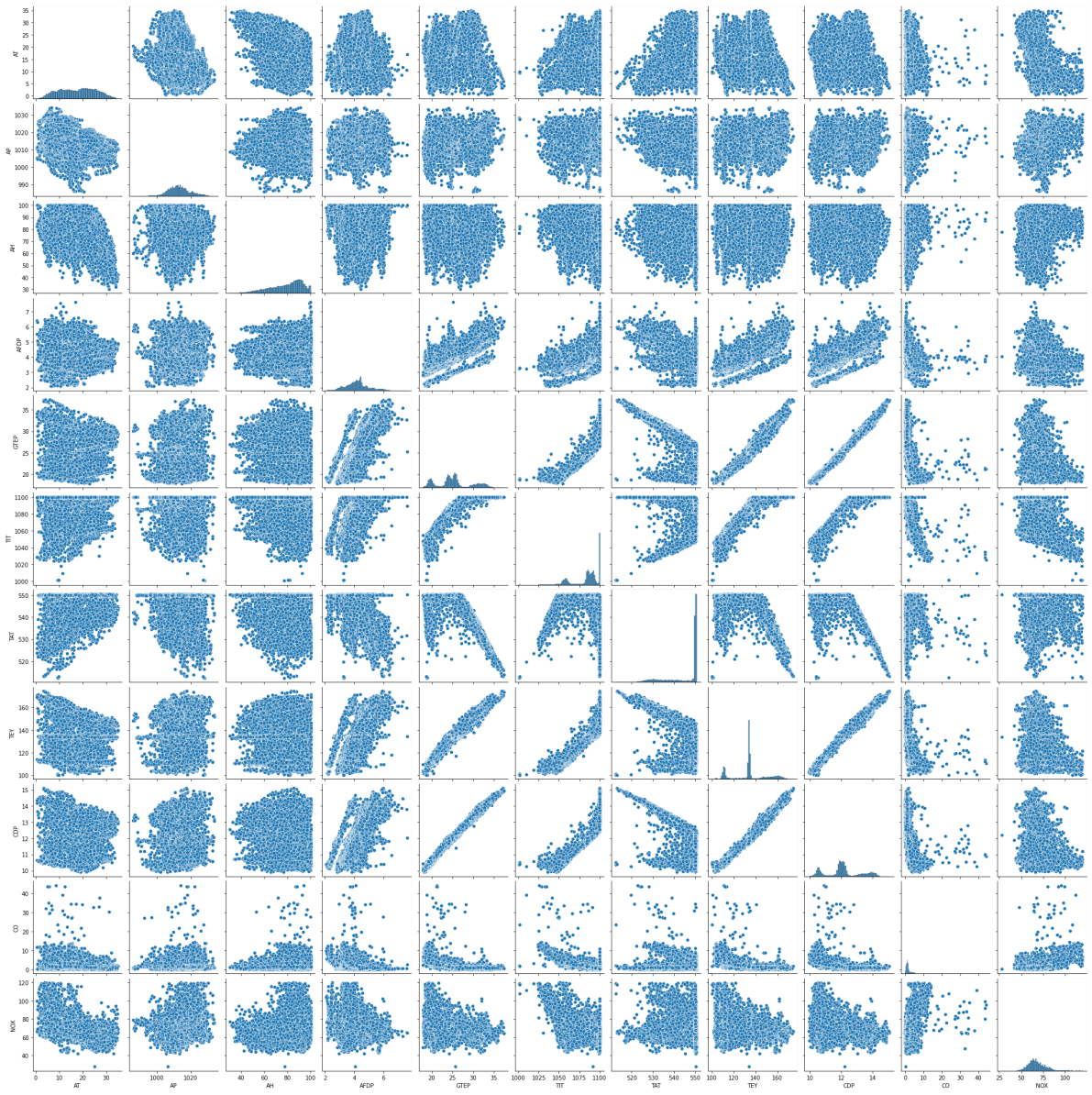
	AT	AP	AH	AFDP	GTEP	TIT	
count	15039.000000	15039.00000	15039.000000	15039.000000	15039.000000	15039.000000	150
mean	17.764381	1013.19924	79.124174	4.200294	25.419061	1083.798770	5
std	7.574323	6.41076	13.793439	0.760197	4.173916	16.527806	
min	0.522300	985.85000	30.344000	2.087400	17.878000	1000.800000	5
25%	11.408000	1008.90000	69.750000	3.723900	23.294000	1079.600000	5
50%	18.186000	1012.80000	82.266000	4.186200	25.082000	1088.700000	5
75%	23.862500	1016.90000	90.043500	4.550900	27.184000	1096.000000	5
max	34.929000	1034.20000	100.200000	7.610600	37.402000	1100.800000	5

In [7]:

```
sns.pairplot(gas_turbine)
```

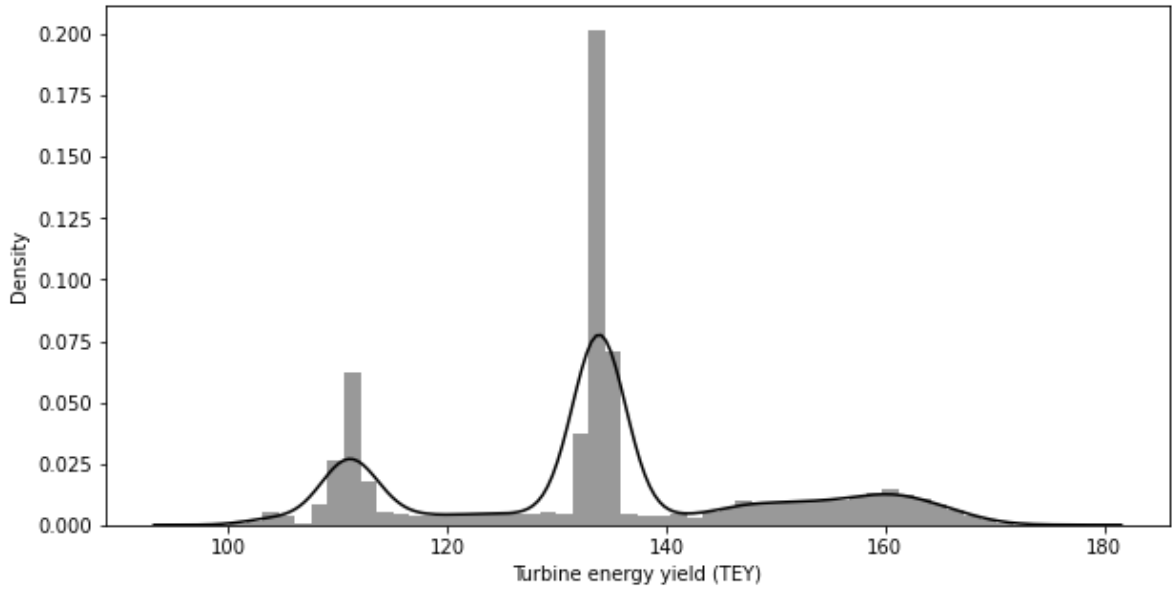
Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x1ba9feb71f0>
```



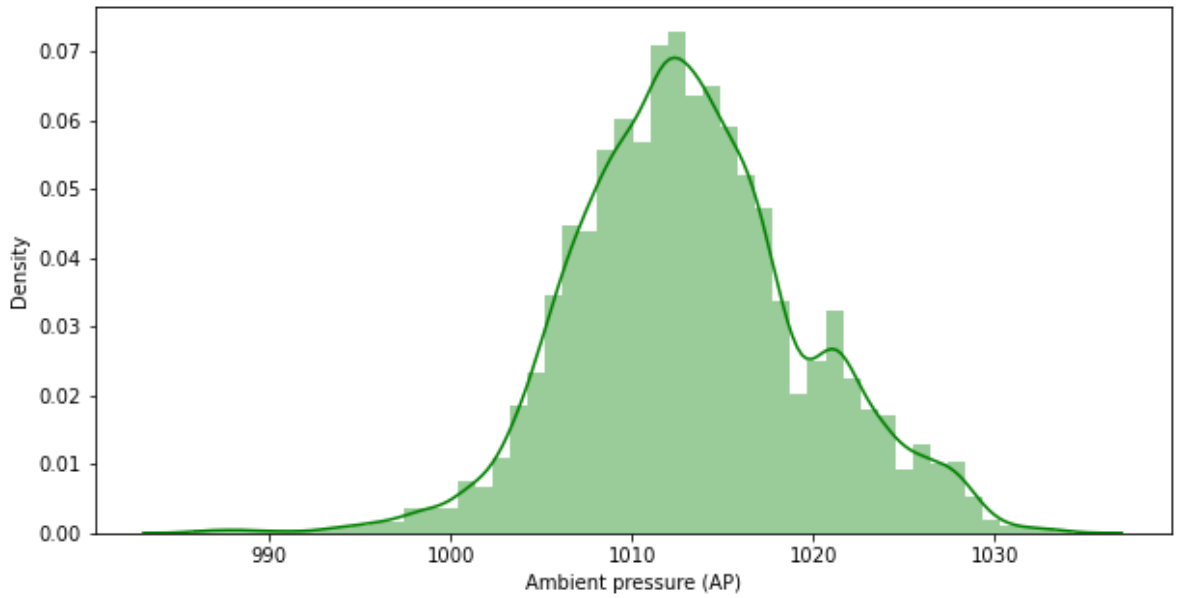
In [8]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['TEY'],axlabel='Turbine energy yield (TEY)',color='black')
plt.show()
```



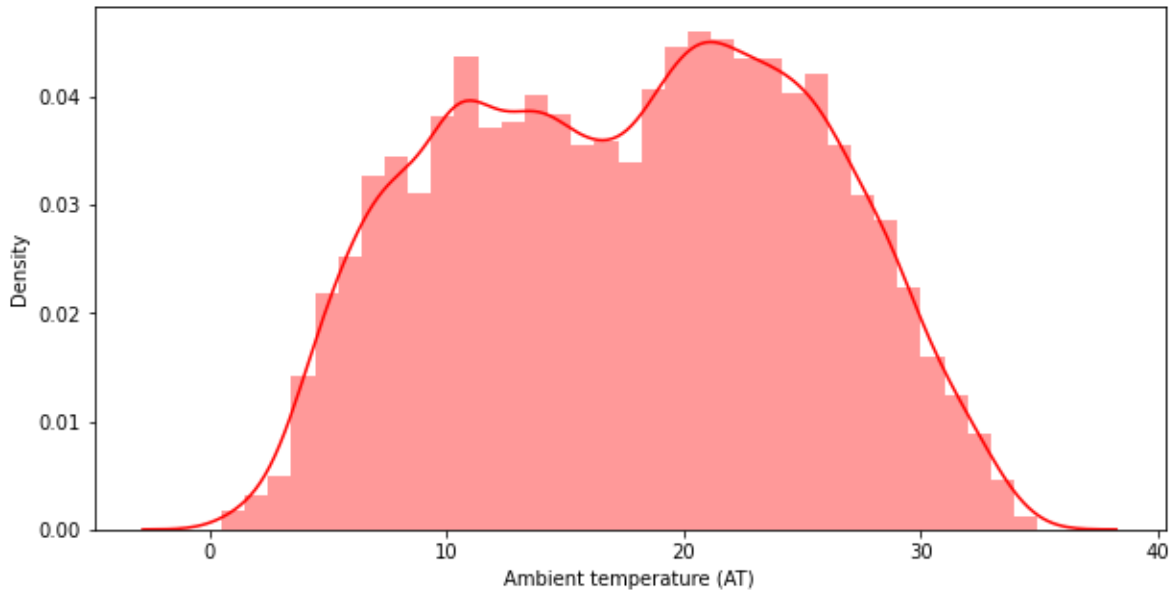
In [9]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['AP'],axlabel='Ambient pressure (AP)',color='green')
plt.show()
```



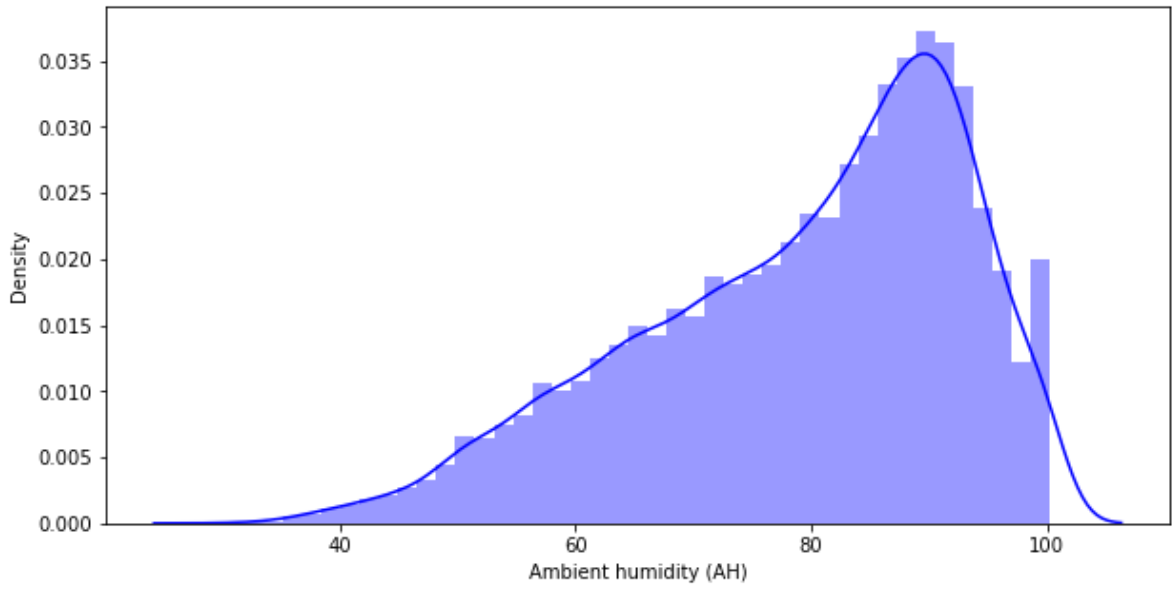
In [10]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['AT'],axlabel='Ambient temperature (AT)',color='red')
plt.show()
```



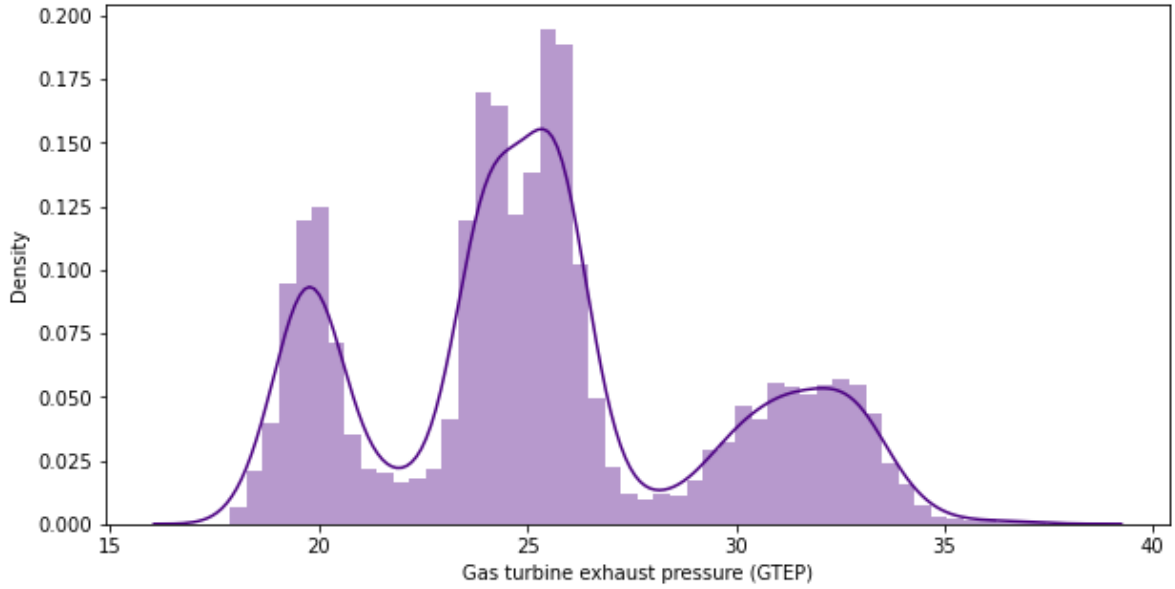
In [11]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['AH'],axlabel='Ambient humidity (AH)',color='blue')
plt.show()
```



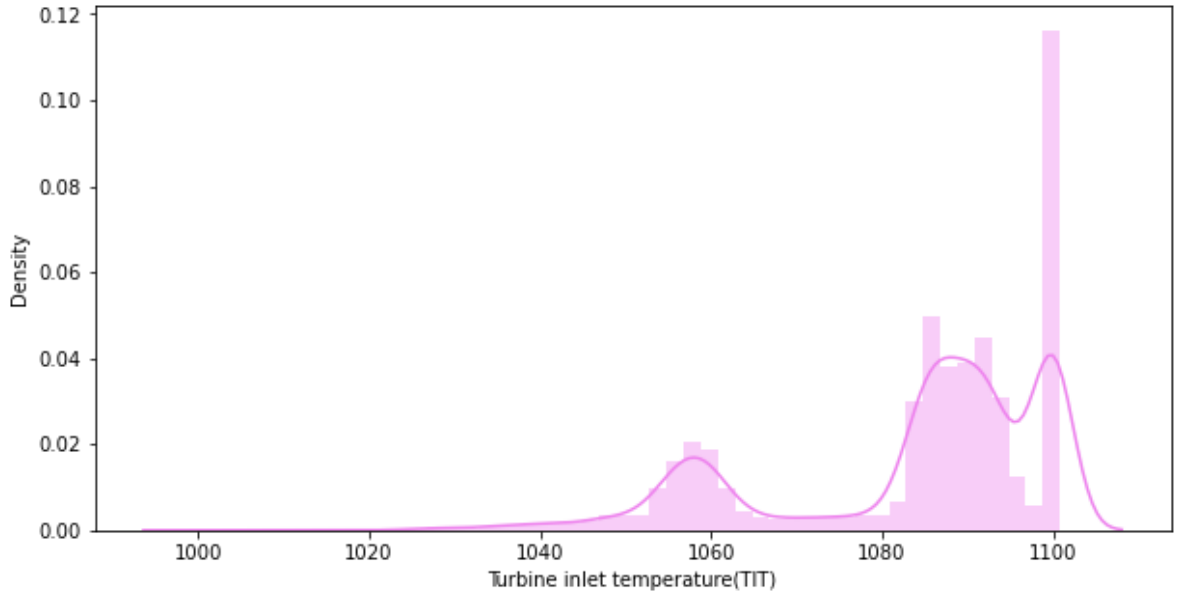
In [12]:

```
#Gas turbine exhaust pressure (GTEP)
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['GTEP'],axlabel='Gas turbine exhaust pressure (GTEP)',color='i
plt.show()
```



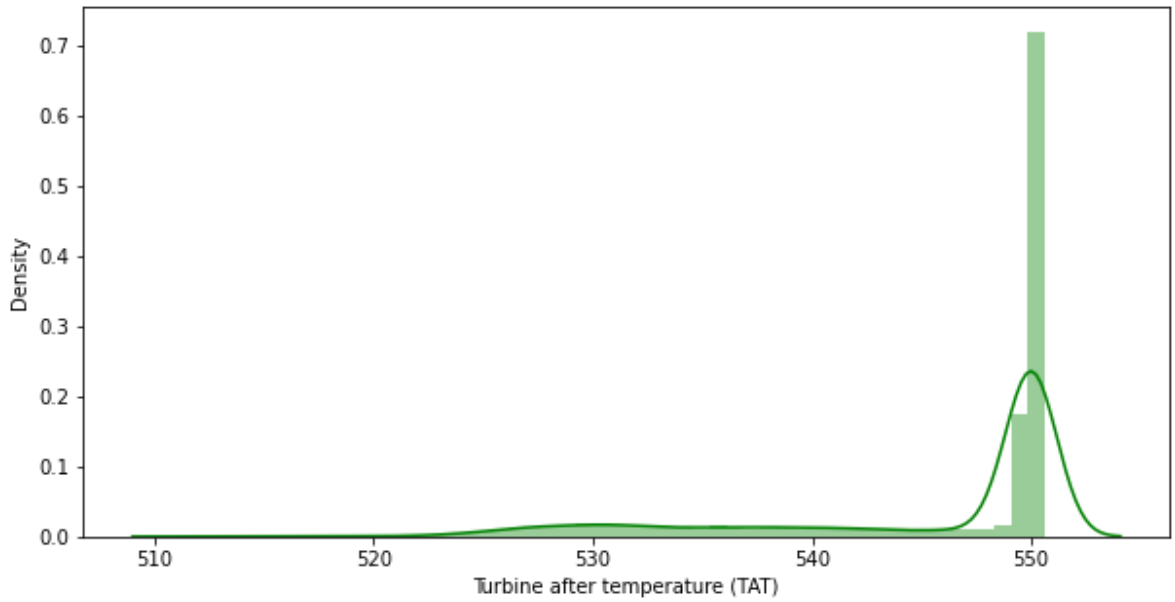
In [13]:

```
#Turbine inlet temperature (TIT)
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['TIT'],axlabel='Turbine inlet temperature(TIT)',color='violet'
plt.show()
```



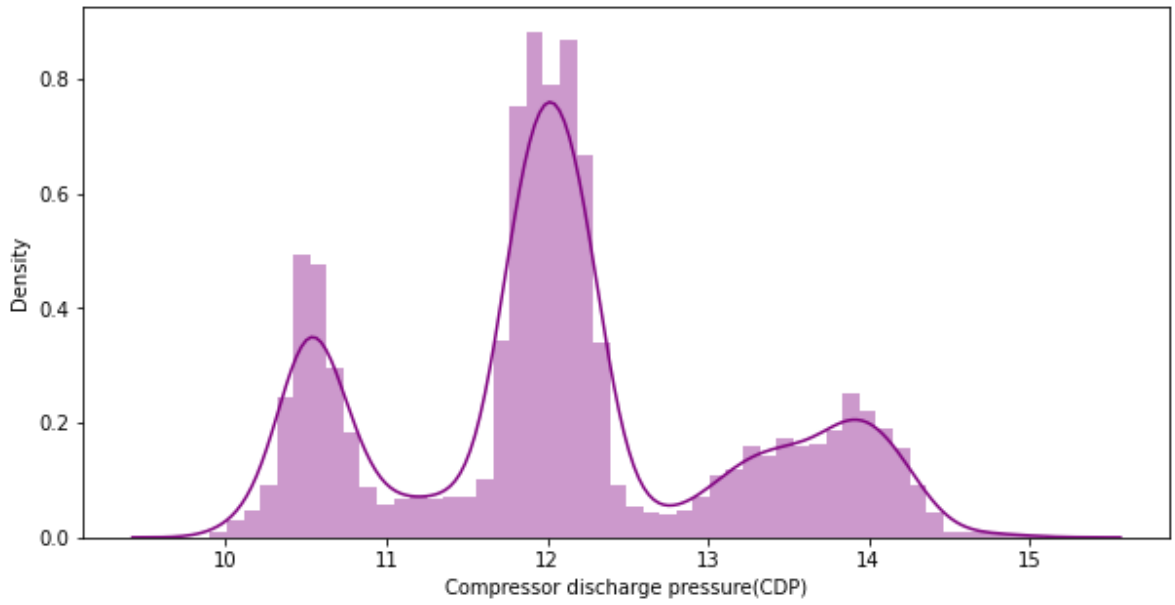
In [14]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['TAT'],axlabel='Turbine after temperature (TAT)',color='green'
plt.show()
```



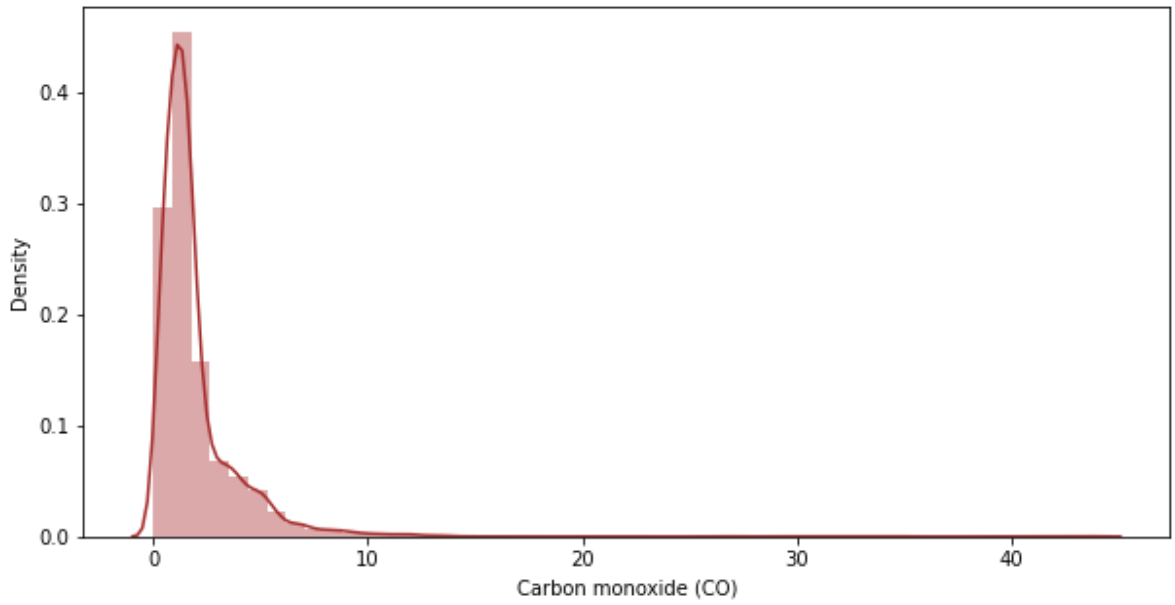
In [15]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['CDP'],axlabel='Compressor discharge pressure(CDP)',color='pur
plt.show()
```



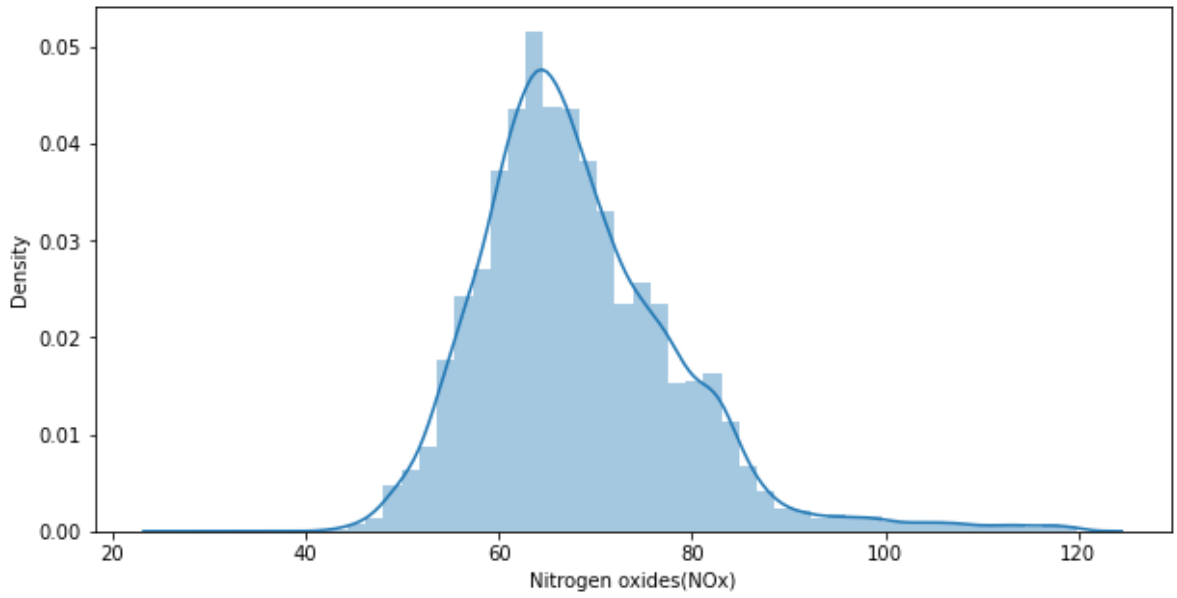
In [16]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['CO'],axlabel='Carbon monoxide (CO)',color='brown')
plt.show()
```



In [17]:

```
plt.figure(figsize=(10,5))
sns.distplot(x = gas_turbine['NOX'],axlabel='Nitrogen oxides(NOx)')
plt.show()
```



In [18]:

```
x = gas_turbine.drop(labels='TEY',axis=1)
y = gas_turbine[['TEY']]
```

In [19]:

```
x
```

Out[19]:

	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 [20]:

```
y
```

Out[20]:

	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 [21]:

```
#Covert input data in standard form
sclar = MinMaxScaler()
scale_data = sclar.fit_transform(x)
scale_data
```

Out[21]:

```
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 [22]:

```
x_train,x_test, y_train, y_test = train_test_split(scale_data,y,test_size=0.30,random_state
```

In [23]:

```
x_train.shape,y_train.shape
```

Out[23]:

```
((10527, 10), (10527, 1))
```

In [24]:

```
x_test.shape,y_test.shape
```

Out[24]:

```
((4512, 10), (4512, 1))
```

In [46]:

```
#Turning Hyperparameter: Batch Size and epochs
def regression_model():
    model = Sequential()
    model.add(Dense(12, input_dim=10,kernel_initializer='uniform',activation='relu'))
    model.add(Dropout(0.2))
    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 [26]:

```
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=5)
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=  2
5.2s
[CV 2/5] END .....batch_size=10, epochs=40;, score=-17884.646 total time=  2
7.0s
[CV 3/5] END .....batch_size=10, epochs=40;, score=-17947.232 total time=  2
8.9s
[CV 4/5] END .....batch_size=10, epochs=40;, score=-18139.010 total time=  2
5.6s
[CV 5/5] END .....batch_size=10, epochs=40;, score=-18041.299 total time=  2
7.7s
[CV 1/5] END .....batch_size=10, epochs=70;, score=-17983.838 total time=  4
2.2s
[CV 2/5] END .....batch_size=10, epochs=70;, score=-17884.646 total time=  4
8.2s
[CV 3/5] END .....batch_size=10, epochs=70;, score=-17947.232 total time=  4
2.2s
[CV 4/5] END .....batch_size=10, epochs=70;, score=-18139.010 total time=  4
5.3s
[CV 5/5] END .....batch_size=10, epochs=70;, score=-18041.299 total time=  4
2.7s
[CV 1/5] END ....batch_size=10, epochs=100;, score=-17983.838 total time=  5
8.4s
[CV 2/5] END ....batch_size=10, epochs=100;, score=-17884.646 total time=  5
9.2s
[CV 3/5] END ....batch_size=10, epochs=100;, score=-17947.232 total time= 1.
2min
[CV 4/5] END ....batch_size=10, epochs=100;, score=-18139.010 total time= 1.
1min
[CV 5/5] END ....batch_size=10, epochs=100;, score=-18041.299 total time= 1.
1min
[CV 1/5] END .....batch_size=50, epochs=40;, score=-17983.840 total time=
5.8s
[CV 2/5] END .....batch_size=50, epochs=40;, score=-17884.643 total time=
5.6s
[CV 3/5] END .....batch_size=50, epochs=40;, score=-17947.230 total time=
7.0s
[CV 4/5] END .....batch_size=50, epochs=40;, score=-18139.010 total time=
6.1s
[CV 5/5] END .....batch_size=50, epochs=40;, score=-18041.297 total time=
6.2s
[CV 1/5] END .....batch_size=50, epochs=70;, score=-17983.840 total time=  1
0.7s
[CV 2/5] END .....batch_size=50, epochs=70;, score=-17884.643 total time=  1
0.4s
[CV 3/5] END .....batch_size=50, epochs=70;, score=-17947.230 total time=  1
0.3s
[CV 4/5] END .....batch_size=50, epochs=70;, score=-18139.010 total time=  1
0.2s
[CV 5/5] END .....batch_size=50, epochs=70;, score=-18041.297 total time=  1
0.3s
```

```
[CV 1/5] END ....batch_size=50, epochs=100;, score=-17983.840 total time=  1
6.1s
[CV 2/5] END ....batch_size=50, epochs=100;, score=-17884.643 total time=  1
4.7s
[CV 3/5] END ....batch_size=50, epochs=100;, score=-17947.230 total time=  1
4.3s
[CV 4/5] END ....batch_size=50, epochs=100;, score=-18139.010 total time=  1
4.8s
[CV 5/5] END ....batch_size=50, epochs=100;, score=-18041.297 total time=  1
5.3s
[CV 1/5] END ....batch_size=100, epochs=40;, score=-17983.834 total time=
3.6s
[CV 2/5] END ....batch_size=100, epochs=40;, score=-17884.643 total time=
3.6s
[CV 3/5] END ....batch_size=100, epochs=40;, score=-17947.230 total time=
3.7s
[CV 4/5] END ....batch_size=100, epochs=40;, score=-18139.010 total time=
3.7s
[CV 5/5] END ....batch_size=100, epochs=40;, score=-18041.297 total time=
3.8s
[CV 1/5] END ....batch_size=100, epochs=70;, score=-17983.834 total time=
5.4s
[CV 2/5] END ....batch_size=100, epochs=70;, score=-17884.643 total time=
5.3s
[CV 3/5] END ....batch_size=100, epochs=70;, score=-17947.230 total time=
5.2s
[CV 4/5] END ....batch_size=100, epochs=70;, score=-18139.010 total time=
5.3s
[CV 5/5] END ....batch_size=100, epochs=70;, score=-18041.297 total time=
5.1s
[CV 1/5] END ...batch_size=100, epochs=100;, score=-17983.834 total time=
7.5s
[CV 2/5] END ...batch_size=100, epochs=100;, score=-17884.643 total time=
7.6s
[CV 3/5] END ...batch_size=100, epochs=100;, score=-17947.230 total time=
7.3s
[CV 4/5] END ...batch_size=100, epochs=100;, score=-18139.010 total time=
7.3s
[CV 5/5] END ...batch_size=100, epochs=100;, score=-18041.297 total time=
7.2s
```

In [27]:

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

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

In [28]:

```
def regression_model(learning_rate,dropout_rate):
    model = Sequential()
    model.add(Dense(12, input_dim=10,kernel_initializer='uniform',activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(8 ,kernel_initializer='uniform',activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(1,kernel_initializer = 'uniform',activation = 'sigmoid'))
    adam = Adam(learning_rate=learning_rate)
    model.compile(loss='mean_squared_error',optimizer=adam, metrics=['mae','mse'])
    return model
```

In [29]:

```
model = KerasRegressor(build_fn=regression_model,batch_size = 100,epochs = 40,verbose = 0)
learning_rate = [0.1,0.01,0.001]
dropout_rate = [0.0,0.1,0.2]
param_grid = dict(learning_rate = learning_rate,dropout_rate = dropout_rate)
gsv = GridSearchCV(estimator= model, param_grid=param_grid, cv=KFold() ,verbose=5)
grid_res = gsv.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
[CV 1/5] END dropout_rate=0.0, learning_rate=0.1;, score=-17983.834 total ti
me=   3.1s
[CV 2/5] END dropout_rate=0.0, learning_rate=0.1;, score=-17884.643 total ti
me=   3.1s
[CV 3/5] END dropout_rate=0.0, learning_rate=0.1;, score=-17947.230 total ti
me=   3.9s
[CV 4/5] END dropout_rate=0.0, learning_rate=0.1;, score=-18139.010 total ti
me=   3.3s
[CV 5/5] END dropout_rate=0.0, learning_rate=0.1;, score=-18041.297 total ti
me=   3.4s
[CV 1/5] END dropout_rate=0.0, learning_rate=0.01;, score=-17983.834 total t
ime=   3.5s
[CV 2/5] END dropout_rate=0.0, learning_rate=0.01;, score=-17884.643 total t
ime=   3.2s
[CV 3/5] END dropout_rate=0.0, learning_rate=0.01;, score=-17947.230 total t
ime=   3.3s
[CV 4/5] END dropout_rate=0.0, learning_rate=0.01;, score=-18139.010 total t
ime=   3.4s
[CV 5/5] END dropout_rate=0.0, learning_rate=0.01;, score=-18041.297 total t
ime=   3.3s
[CV 1/5] END dropout_rate=0.0, learning_rate=0.001;, score=-17983.834 total
time=   3.1s
[CV 2/5] END dropout_rate=0.0, learning_rate=0.001;, score=-17884.643 total
time=   3.2s
[CV 3/5] END dropout_rate=0.0, learning_rate=0.001;, score=-17947.230 total
time=   3.3s
[CV 4/5] END dropout_rate=0.0, learning_rate=0.001;, score=-18139.010 total
time=   3.2s
[CV 5/5] END dropout_rate=0.0, learning_rate=0.001;, score=-18041.297 total
time=   3.2s
[CV 1/5] END dropout_rate=0.1, learning_rate=0.1;, score=-17983.834 total ti
me=   3.3s
[CV 2/5] END dropout_rate=0.1, learning_rate=0.1;, score=-17884.643 total ti
me=   3.3s
[CV 3/5] END dropout_rate=0.1, learning_rate=0.1;, score=-17947.230 total ti
me=   3.4s
[CV 4/5] END dropout_rate=0.1, learning_rate=0.1;, score=-18139.010 total ti
me=   3.2s
[CV 5/5] END dropout_rate=0.1, learning_rate=0.1;, score=-18041.297 total ti
me=   3.3s
[CV 1/5] END dropout_rate=0.1, learning_rate=0.01;, score=-17983.834 total t
ime=   3.3s
[CV 2/5] END dropout_rate=0.1, learning_rate=0.01;, score=-17884.643 total t
ime=   3.4s
[CV 3/5] END dropout_rate=0.1, learning_rate=0.01;, score=-17947.230 total t
ime=   4.0s
[CV 4/5] END dropout_rate=0.1, learning_rate=0.01;, score=-18139.010 total t
ime=   3.5s
[CV 5/5] END dropout_rate=0.1, learning_rate=0.01;, score=-18041.297 total t
ime=   3.9s
```

```
[CV 1/5] END dropout_rate=0.1, learning_rate=0.001;, score=-17983.834 total
time=   3.3s
[CV 2/5] END dropout_rate=0.1, learning_rate=0.001;, score=-17884.643 total
time=   3.6s
[CV 3/5] END dropout_rate=0.1, learning_rate=0.001;, score=-17947.230 total
time=   3.7s
[CV 4/5] END dropout_rate=0.1, learning_rate=0.001;, score=-18139.010 total
time=   3.5s
[CV 5/5] END dropout_rate=0.1, learning_rate=0.001;, score=-18041.297 total
time=   3.3s
[CV 1/5] END dropout_rate=0.2, learning_rate=0.1;, score=-17983.834 total ti
me=   3.4s
[CV 2/5] END dropout_rate=0.2, learning_rate=0.1;, score=-17884.643 total ti
me=   3.3s
[CV 3/5] END dropout_rate=0.2, learning_rate=0.1;, score=-17947.230 total ti
me=   3.4s
[CV 4/5] END dropout_rate=0.2, learning_rate=0.1;, score=-18139.010 total ti
me=   3.3s
[CV 5/5] END dropout_rate=0.2, learning_rate=0.1;, score=-18041.297 total ti
me=   3.3s
[CV 1/5] END dropout_rate=0.2, learning_rate=0.01;, score=-17983.834 total t
ime=   3.4s
[CV 2/5] END dropout_rate=0.2, learning_rate=0.01;, score=-17884.643 total t
ime=   3.3s
[CV 3/5] END dropout_rate=0.2, learning_rate=0.01;, score=-17947.230 total t
ime=   3.2s
[CV 4/5] END dropout_rate=0.2, learning_rate=0.01;, score=-18139.010 total t
ime=   3.3s
[CV 5/5] END dropout_rate=0.2, learning_rate=0.01;, score=-18041.297 total t
ime=   3.6s
[CV 1/5] END dropout_rate=0.2, learning_rate=0.001;, score=-17983.834 total
time=   4.0s
[CV 2/5] END dropout_rate=0.2, learning_rate=0.001;, score=-17884.643 total
time=   3.4s
[CV 3/5] END dropout_rate=0.2, learning_rate=0.001;, score=-17947.230 total
time=   3.3s
[CV 4/5] END dropout_rate=0.2, learning_rate=0.001;, score=-18139.010 total
time=   3.2s
[CV 5/5] END dropout_rate=0.2, learning_rate=0.001;, score=-18041.297 total
time=   3.1s
```

In [30]:

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

```
-17999.202734375 {'dropout_rate': 0.0, 'learning_rate': 0.1}
```

In [31]:

```
def regression_model(activation_function,inti):
    model = Sequential()
    model.add(Dense(12, input_dim=10,kernel_initializer='uniform',activation='relu'))
    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,verbose = 0)
activation_function = ['relu', 'tanh','softmax','linear']
inti = ['uniform', 'zero','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=uniform;, score=-17983.834 tot
al time=   3.0s
[CV 2/5] END activation_function=relu, inti=uniform;, score=-17884.643 tot
al time=   2.9s
[CV 3/5] END activation_function=relu, inti=uniform;, score=-17947.230 tot
al time=   3.5s
[CV 4/5] END activation_function=relu, inti=uniform;, score=-18139.010 tot
al time=   3.1s
[CV 5/5] END activation_function=relu, inti=uniform;, score=-18041.297 tot
al time=   3.0s
[CV 1/5] END activation_function=relu, inti=zero;, score=-17983.834 total
time=   3.0s
[CV 2/5] END activation_function=relu, inti=zero;, score=-17884.643 total
time=   3.0s
[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=   3.5s
[CV 5/5] END activation_function=relu, inti=zero;, score=-18041.297 total
time=   3.5s
[CV 1/5] END activation_function=relu, inti=normal;, score=-17983.834 tota
l time=   3.7s
[CV 2/5] END activation_function=relu, inti=normal;, score=-17884.643 tota
l time=   3.1s
[CV 3/5] END activation_function=relu, inti=normal;, score=-17947.230 tota
l time=   2.7s
[CV 4/5] END activation_function=relu, inti=normal;, score=-18139.010 tota
l time=   2.9s
[CV 5/5] END activation_function=relu, inti=normal;, score=-18041.297 tota
l time=   3.1s
[CV 1/5] END activation_function=tanh, inti=uniform;, score=-17983.834 tot
al time=   2.8s
[CV 2/5] END activation_function=tanh, inti=uniform;, score=-17884.643 tot
al time=   4.7s
[CV 3/5] END activation_function=tanh, inti=uniform;, score=-17947.230 tot
al time=   3.7s
[CV 4/5] END activation_function=tanh, inti=uniform;, score=-18139.010 tot
al time=   3.0s
[CV 5/5] END activation_function=tanh, inti=uniform;, score=-18041.297 tot
al time=   3.0s
[CV 1/5] END activation_function=tanh, inti=zero;, score=-17983.834 total
time=   3.1s
[CV 2/5] END activation_function=tanh, inti=zero;, score=-17884.643 total
time=   2.9s
[CV 3/5] END activation_function=tanh, inti=zero;, score=-17947.230 total
time=   2.9s
[CV 4/5] END activation_function=tanh, inti=zero;, score=-18139.010 total
time=   3.1s
[CV 5/5] END activation_function=tanh, inti=zero;, score=-18041.297 total
time=   2.8s
[CV 1/5] END activation_function=tanh, inti=normal;, score=-17983.834 tota
```

```
l time=   3.1s
[CV 2/5] END activation_function=tanh, inti=normal;, score=-17884.643 tota
l time=   3.7s
[CV 3/5] END activation_function=tanh, inti=normal;, score=-17947.230 tota
l time=   3.4s
[CV 4/5] END activation_function=tanh, inti=normal;, score=-18139.010 tota
l time=   3.4s
[CV 5/5] END activation_function=tanh, inti=normal;, score=-18041.297 tota
l time=   4.1s
[CV 1/5] END activation_function=softmax, inti=uniform;, score=-17983.834
total time=   4.2s
[CV 2/5] END activation_function=softmax, inti=uniform;, score=-17884.643
total time=   3.1s
[CV 3/5] END activation_function=softmax, inti=uniform;, score=-17947.230
total time=   3.6s
[CV 4/5] END activation_function=softmax, inti=uniform;, score=-18139.010
total time=   3.3s
[CV 5/5] END activation_function=softmax, inti=uniform;, score=-18041.297
total time=   3.6s
[CV 1/5] END activation_function=softmax, inti=zero;, score=-17983.834 tot
al time=   3.4s
[CV 2/5] END activation_function=softmax, inti=zero;, score=-17884.643 tot
al time=   3.3s
[CV 3/5] END activation_function=softmax, inti=zero;, score=-17947.230 tot
al time=   3.3s
[CV 4/5] END activation_function=softmax, inti=zero;, score=-18139.010 tot
al time=   3.5s
[CV 5/5] END activation_function=softmax, inti=zero;, score=-18041.297 tot
al time=   3.4s
[CV 1/5] END activation_function=softmax, inti=normal;, score=-17983.834 t
otal time=   3.4s
[CV 2/5] END activation_function=softmax, inti=normal;, score=-17884.643 t
otal time=   3.2s
[CV 3/5] END activation_function=softmax, inti=normal;, score=-17947.230 t
otal time=   3.2s
[CV 4/5] END activation_function=softmax, inti=normal;, score=-18139.010 t
otal time=   3.0s
[CV 5/5] END activation_function=softmax, inti=normal;, score=-18041.297 t
otal time=   3.1s
[CV 1/5] END activation_function=linear, inti=uniform;, score=-17983.834 t
otal time=   3.7s
[CV 2/5] END activation_function=linear, inti=uniform;, score=-17884.643 t
otal time=   3.1s
[CV 3/5] END activation_function=linear, inti=uniform;, score=-17947.230 t
otal time=   2.8s
[CV 4/5] END activation_function=linear, inti=uniform;, score=-18139.010 t
otal time=   2.9s
[CV 5/5] END activation_function=linear, inti=uniform;, score=-18041.297 t
otal time=   3.2s
[CV 1/5] END activation_function=linear, inti=zero;, score=-17983.834 tota
l time=   2.9s
[CV 2/5] END activation_function=linear, inti=zero;, score=-17884.643 tota
l time=   2.9s
[CV 3/5] END activation_function=linear, inti=zero;, score=-17947.230 tota
l time=   3.5s
[CV 4/5] END activation_function=linear, inti=zero;, score=-18139.010 tota
l time=   3.1s
[CV 5/5] END activation_function=linear, inti=zero;, score=-18041.297 tota
l time=   3.0s
[CV 1/5] END activation_function=linear, inti=normal;, score=-17983.834 to
tal time=   3.2s
```

```
[CV 2/5] END activation_function=linear, inti=normal;, score=-17884.643 to
tal time=   3.2s
[CV 3/5] END activation_function=linear, inti=normal;, score=-17947.230 to
tal time=   3.2s
[CV 4/5] END activation_function=linear, inti=normal;, score=-18139.010 to
tal time=   3.2s
[CV 5/5] END activation_function=linear, inti=normal;, score=-18041.297 to
tal time=   3.1s
```

In [33]:

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

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

In [34]:

```
def regression_model(neuron1,neuron2):
    model = Sequential()
    model.add(Dense(12,input_dim = 10,kernel_initializer='uniform',activation='relu'))
    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,verbose = 0)
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=5)
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=
3.3s
[CV 2/5] END .......neuron1=16, neuron2=12;, score=-17884.643 total time=
3.2s
[CV 3/5] END .......neuron1=16, neuron2=12;, score=-17947.230 total time=
3.2s
[CV 4/5] END .......neuron1=16, neuron2=12;, score=-18139.010 total time=
3.8s
[CV 5/5] END .......neuron1=16, neuron2=12;, score=-18041.297 total time=
3.5s
[CV 1/5] END ........neuron1=16, neuron2=8;, score=-17983.834 total time=
3.4s
[CV 2/5] END ........neuron1=16, neuron2=8;, score=-17884.643 total time=
3.5s
[CV 3/5] END ........neuron1=16, neuron2=8;, score=-17947.230 total time=
3.2s
[CV 4/5] END ........neuron1=16, neuron2=8;, score=-18139.010 total time=
3.0s
[CV 5/5] END ........neuron1=16, neuron2=8;, score=-18041.297 total time=
3.2s
[CV 1/5] END ........neuron1=16, neuron2=4;, score=-17983.834 total time=
2.9s
[CV 2/5] END ........neuron1=16, neuron2=4;, score=-17884.643 total time=
2.8s
[CV 3/5] END ........neuron1=16, neuron2=4;, score=-17947.230 total time=
2.8s
[CV 4/5] END ........neuron1=16, neuron2=4;, score=-18139.010 total time=
3.3s
[CV 5/5] END ........neuron1=16, neuron2=4;, score=-18041.297 total time=
3.3s
[CV 1/5] END .......neuron1=12, neuron2=12;, score=-17983.834 total time=
3.0s
[CV 2/5] END .......neuron1=12, neuron2=12;, score=-17884.654 total time=
2.9s
[CV 3/5] END .......neuron1=12, neuron2=12;, score=-17947.230 total time=
3.4s
[CV 4/5] END .......neuron1=12, neuron2=12;, score=-18139.010 total time=
3.3s
[CV 5/5] END .......neuron1=12, neuron2=12;, score=-18041.297 total time=
3.4s
[CV 1/5] END ........neuron1=12, neuron2=8;, score=-17983.834 total time=
3.3s
[CV 2/5] END ........neuron1=12, neuron2=8;, score=-17884.643 total time=
4.1s
[CV 3/5] END ........neuron1=12, neuron2=8;, score=-17947.230 total time=
3.3s
[CV 4/5] END ........neuron1=12, neuron2=8;, score=-18139.010 total time=
3.2s
[CV 5/5] END ........neuron1=12, neuron2=8;, score=-18041.297 total time=
3.8s
[CV 1/5] END ........neuron1=12, neuron2=4;, score=-17983.834 total time=
```

```
3.2s
[CV 2/5] END ........neuron1=12, neuron2=4;, score=-17884.643 total time=
3.3s
[CV 3/5] END ........neuron1=12, neuron2=4;, score=-17947.230 total time=
3.3s
[CV 4/5] END ........neuron1=12, neuron2=4;, score=-18139.010 total time=
3.2s
[CV 5/5] END ........neuron1=12, neuron2=4;, score=-18041.297 total time=
3.1s
[CV 1/5] END ........neuron1=8, neuron2=12;, score=-17983.834 total time=
3.1s
[CV 2/5] END ........neuron1=8, neuron2=12;, score=-17884.643 total time=
3.0s
[CV 3/5] END ........neuron1=8, neuron2=12;, score=-17947.230 total time=
3.1s
[CV 4/5] END ........neuron1=8, neuron2=12;, score=-18139.010 total time=
2.8s
[CV 5/5] END ........neuron1=8, neuron2=12;, score=-18041.297 total time=
3.1s
[CV 1/5] END .........neuron1=8, neuron2=8;, score=-17983.834 total time=
3.0s
[CV 2/5] END .........neuron1=8, neuron2=8;, score=-17884.643 total time=
3.1s
[CV 3/5] END .........neuron1=8, neuron2=8;, score=-17947.230 total time=
3.2s
[CV 4/5] END .........neuron1=8, neuron2=8;, score=-18139.010 total time=
3.0s
[CV 5/5] END .........neuron1=8, neuron2=8;, score=-18041.297 total time=
3.0s
[CV 1/5] END .........neuron1=8, neuron2=4;, score=-17983.834 total time=
2.9s
[CV 2/5] END .........neuron1=8, neuron2=4;, score=-17884.643 total time=
3.0s
[CV 3/5] END .........neuron1=8, neuron2=4;, score=-17947.230 total time=
3.6s
[CV 4/5] END .........neuron1=8, neuron2=4;, score=-18139.010 total time=
4.4s
[CV 5/5] END .........neuron1=8, neuron2=4;, score=-18041.297 total time=
3.8s
```

In [36]:

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

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

In [37]:

```
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 [38]:

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

```
Epoch 1/40
106/106 [==============================] - 0s 887us/step - loss: 1145.5724
- mae: 27.4509 - mse: 1145.5724
Epoch 2/40
106/106 [==============================] - 0s 1ms/step - loss: 644.2322 -
mae: 24.5131 - mse: 644.2322
Epoch 3/40
106/106 [==============================] - 0s 2ms/step - loss: 508.5970 -
mae: 21.7649 - mse: 508.5970
Epoch 4/40
106/106 [==============================] - 0s 2ms/step - loss: 411.6340 -
mae: 19.5252 - mse: 411.6339
Epoch 5/40
106/106 [==============================] - 0s 2ms/step - loss: 330.0581 -
mae: 17.5133 - mse: 330.0581
Epoch 6/40
106/106 [==============================] - 0s 2ms/step - loss: 267.2840 -
mae: 15.7131 - mse: 267.2840
Epoch 7/40
106/106 [==============================] - 0s 2ms/step - loss: 203.3139 -
mae: 13.6319 - mse: 203.3139
Epoch 8/40
106/106 [==============================] - 0s 2ms/step - loss: 150.8264 -
mae: 11.6620 - mse: 150.8264
Epoch 9/40
106/106 [==============================] - 0s 1ms/step - loss: 143.3845 -
mae: 11.4562 - mse: 143.3845
Epoch 10/40
106/106 [==============================] - 0s 1ms/step - loss: 132.8820 -
mae: 11.0877 - mse: 132.8820
Epoch 11/40
106/106 [==============================] - 0s 1ms/step - loss: 117.6851 -
mae: 10.3730 - mse: 117.6851
Epoch 12/40
106/106 [==============================] - 0s 1ms/step - loss: 88.4665 - m
ae: 8.9199 - mse: 88.4665
Epoch 13/40
106/106 [==============================] - 0s 1ms/step - loss: 83.7524 - m
ae: 8.7815 - mse: 83.7524
Epoch 14/40
106/106 [==============================] - 0s 1ms/step - loss: 78.6902 - m
ae: 8.5520 - mse: 78.6902
Epoch 15/40
106/106 [==============================] - 0s 1ms/step - loss: 74.5052 - m
ae: 8.3357 - mse: 74.5052
Epoch 16/40
106/106 [==============================] - 0s 1ms/step - loss: 72.0202 - m
ae: 8.2237 - mse: 72.0202
Epoch 17/40
106/106 [==============================] - 0s 1ms/step - loss: 67.8682 - m
ae: 7.9936 - mse: 67.8682
Epoch 18/40
106/106 [==============================] - 0s 1ms/step - loss: 64.4158 - m
ae: 7.7921 - mse: 64.4158
Epoch 19/40
```

```
106/106 [==============================] - 0s 1ms/step - loss: 61.2085 - m
ae: 7.6056 - mse: 61.2085
Epoch 20/40
106/106 [==============================] - 0s 1ms/step - loss: 56.9296 - m
ae: 7.3204 - mse: 56.9296
Epoch 21/40
106/106 [==============================] - 0s 997us/step - loss: 52.3595 -
mae: 6.9827 - mse: 52.3595
Epoch 22/40
106/106 [==============================] - 0s 1ms/step - loss: 48.7749 - m
ae: 6.6801 - mse: 48.7749
Epoch 23/40
106/106 [==============================] - 0s 987us/step - loss: 43.6461 -
mae: 6.1847 - mse: 43.6461
Epoch 24/40
106/106 [==============================] - 0s 907us/step - loss: 38.8014 -
mae: 5.7874 - mse: 38.8014
Epoch 25/40
106/106 [==============================] - 0s 1ms/step - loss: 33.3175 - m
ae: 5.2309 - mse: 33.3175
Epoch 26/40
106/106 [==============================] - 0s 1ms/step - loss: 31.6502 - m
ae: 5.0252 - mse: 31.6502
Epoch 27/40
106/106 [==============================] - 0s 1ms/step - loss: 29.9286 - m
ae: 4.7468 - mse: 29.9286
Epoch 28/40
106/106 [==============================] - 0s 1ms/step - loss: 30.2190 - m
ae: 4.8082 - mse: 30.2190
Epoch 29/40
106/106 [==============================] - 0s 1ms/step - loss: 29.8657 - m
ae: 4.8139 - mse: 29.8657
Epoch 30/40
106/106 [==============================] - 0s 1ms/step - loss: 29.5983 - m
ae: 4.6811 - mse: 29.5983
Epoch 31/40
106/106 [==============================] - 0s 1ms/step - loss: 28.5024 - m
ae: 4.5621 - mse: 28.5024
Epoch 32/40
106/106 [==============================] - 0s 1ms/step - loss: 29.5072 - m
ae: 4.7378 - mse: 29.5072
Epoch 33/40
106/106 [==============================] - 0s 924us/step - loss: 28.6395 -
mae: 4.7121 - mse: 28.6395
Epoch 34/40
106/106 [==============================] - 0s 883us/step - loss: 28.1203 -
mae: 4.4385 - mse: 28.1203
Epoch 35/40
106/106 [==============================] - 0s 1ms/step - loss: 27.3642 - m
ae: 4.3786 - mse: 27.3642
Epoch 36/40
106/106 [==============================] - 0s 1ms/step - loss: 28.1318 - m
ae: 4.5152 - mse: 28.1318
Epoch 37/40
106/106 [==============================] - 0s 1ms/step - loss: 28.2070 - m
ae: 4.7405 - mse: 28.2070
Epoch 38/40
106/106 [==============================] - 0s 1ms/step - loss: 26.4947 - m
ae: 4.3369 - mse: 26.4947
Epoch 39/40
106/106 [==============================] - 0s 1ms/step - loss: 28.6497 - m
```

```
ae: 4.4705 - mse: 28.6497
Epoch 40/40
106/106 [==============================] - 0s 929us/step - loss: 26.1394 -
mae: 4.1729 - mse: 26.1394
329/329 [==============================] - 0s 589us/step
```

In [39]:

```
model.summary()
```

```
Model: "sequential_199"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_597 (Dense)           (None, 16)                176

 dropout_398 (Dropout)       (None, 16)                0

 dense_598 (Dense)           (None, 12)                204

 dropout_399 (Dropout)       (None, 12)                0

 dense_599 (Dense)           (None, 1)                 13

=================================================================
Total params: 393
Trainable params: 393
Non-trainable params: 0
_________________________________________________________________
```

In [40]:

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

Out[40]:

```
5.98439602677657
```

In [41]:

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

Out[41]:

```
45.82506077138494
```

In [42]:

```
test_score = model.evaluate(x_test,y_test)
```

```
141/141 [==============================] - 0s 792us/step - loss: 46.0209 - m
ae: 5.9826 - mse: 46.0209
```

In [43]:

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

```
141/141 [==============================] - 0s 730us/step
```

In [44]:

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

Out[44]:

```
5.982551945625468
```

In [45]:

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

Out[45]:

```
46.020875714849154
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```