```
# -*- coding: utf-8 -*-
"""
Created on Sat Mar 28 23:27:04 2020

@author: Admin
"""
#Importing the libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

#Importing the dataset
dataset=pd.read_csv('CLC_train.csv')

X_COgt=dataset.iloc[:,2].values
X_Pt08s1=dataset.iloc[:,3].values
X_Nmhcgt=dataset.iloc[:,4].values
X_C6H6=dataset.iloc[:,5].values
X_Pt08s2=dataset.iloc[:,6].values
X_Nox=dataset.iloc[:,7].values
X_Pt08s3=dataset.iloc[:,8].values
X_No2=dataset.iloc[:,9].values
X_Pt08s4=dataset.iloc[:,10].values
X_Pt08s5=dataset.iloc[:,11].values
X_T=dataset.iloc[:,12].values
X_RH=dataset.iloc[:,13].values
X_AH=dataset.iloc[:,14].values

Y=dataset.iloc[:,15].values

'''from sklearn.preprocessing import LabelEncoder,OneHotEncoder
labelencoder_Y=LabelEncoder()
one=OneHotEncoder(categorical_features=[0])
Y=one.fit_transform(Y).toarray()'''
```

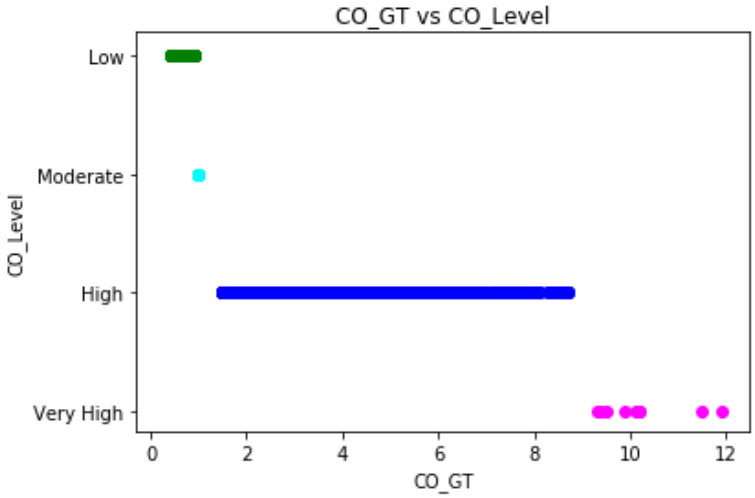
```
#Removing -200 from columns and accordingly change the CO_level column
X_Cogt_=X_COgt[X_COgt!=-200]
X_Pt08s1_=X_Pt08s1[X_Pt08s1!=-200]
X_Nmhcgt_=X_Nmhcgt[X_Nmhcgt!=-200]
X_C6H6_=X_C6H6[X_C6H6!=-200]
X_Pt08s2_=X_Pt08s2[X_Pt08s2!=-200]
X_Nox_=X_Nox[X_Nox!=-200]
X_Pt08s3_=X_Pt08s3[X_Pt08s3!=-200]
X_No2_=X_No2[X_No2!=-200]
X_Pt08s4_=X_Pt08s4[X_Pt08s4!=-200]
X_Pt08s5_=X_Pt08s5[X_Pt08s5!=-200]
X_T_=X_T[X_T!=-200]
X_RH_=X_RH[X_RH!=-200]
X_AH_=X_AH[X_AH!=-200]

a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13=[],[],[],[],[],[],[],[],[],[],[],[],[]
for i in range(0,7485):
if X_COgt[i]==-200:a1.append(i)
if X_Pt08s1[i]==-200:a2.append(i)
if X_Nmhcgt[i]==-200:a3.append(i)
if X_C6H6[i]==-200:a4.append(i)
if X_Pt08s2[i]==-200:a5.append(i)
if X_Nox[i]==-200:a6.append(i)
if X_Pt08s3[i]==-200:a7.append(i)
if X_No2[i]==-200:a8.append(i)
if X_Pt08s4[i]==-200:a9.append(i)
if X_Pt08s5[i]==-200:a10.append(i)
if X_T[i]==-200:a11.append(i)
if X_RH[i]==-200:a12.append(i)
if X_AH[i]==-200:a13.append(i)


Y_Cogt=np.delete(Y,a1)
Y_Pt08s1=np.delete(Y,a2)
Y_Nmhcgt=np.delete(Y,a3)
Y_C6H6=np.delete(Y,a4)
Y_Pt08s2=np.delete(Y,a5)
Y_Nox=np.delete(Y,a6)
Y_Pt08s3=np.delete(Y,a7)
Y_No2=np.delete(Y,a8)
Y_Pt08s4=np.delete(Y,a9)
Y_Pt08s5=np.delete(Y,a10)
Y_T=np.delete(Y,a11)
Y_RH=np.delete(Y,a12)
Y_AH=np.delete(Y,a13)
```

```
colors={'Very low':'red','Low':'green','Moderate':'cyan','High':'blue','Very High':'magenta'}
```

```
#Plotting CO_GT vs CO_level
```

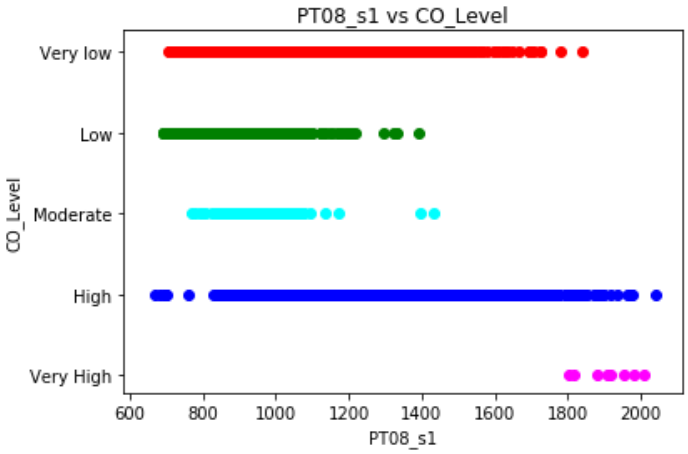
```
#plt.xlim(1250,1500)
for i in range(len(X_Cogt_)):
plt.scatter(X_Cogt_[i],Y_Cogt[i],color=colors[Y_Cogt[i]])
plt.title('CO_GT vs CO_Level')
plt.xlabel('CO_GT')
plt.ylabel('CO_Level')
plt.show()
```

O/P:




```
#Plotting PT08_S1 vs CO_level
```

```
#plt.xlim(1250,1500)
for i in range(len(X_Pt08s1_)):
plt.scatter(X_Pt08s1_[i],Y_Pt08s1[i],color=colors[Y_Pt08s1[i]])
plt.title('PT08_s1 vs CO_Level')
plt.xlabel('PT08_s1')
plt.ylabel('CO_Level')
plt.show()
```

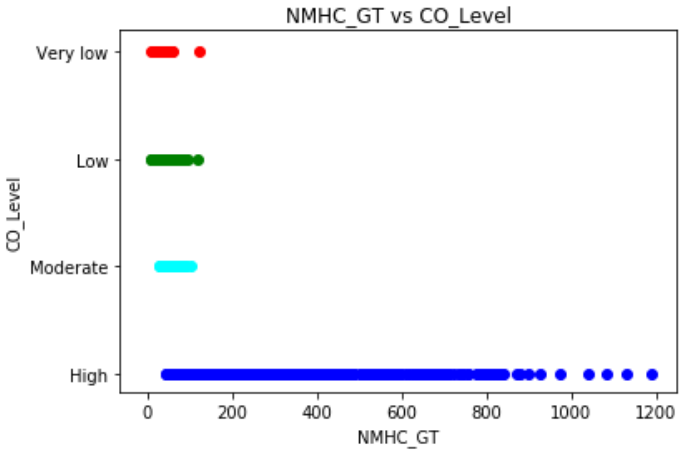


O/P:

89. *#Plotting NMHC_GT vs CO_level*

90. *#plt.xlim(1250,1500)*
91. for i in range(len(X_Nmhcgt_)):
92. plt.scatter(X_Nmhcgt_[i],Y_Nmhcgt[i],color=colors[Y_Nmhcgt[i]])
93. plt.title('NMHC_GT vs CO_Level')
94. plt.xlabel('NMHC_GT')
95. plt.ylabel('CO_Level')
96. plt.show()

O/P:

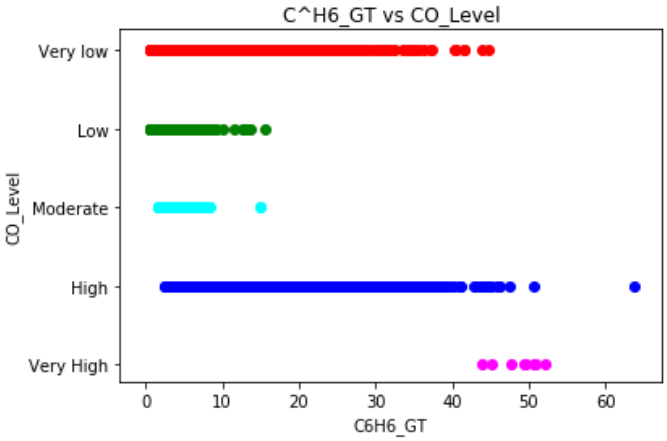


97. *#Plotting C6H6_GT vs CO_level*

98. *#plt.xlim(1250,1500)*
99. for i in range(len(X_C6H6_)):
100. plt.scatter(X_C6H6_[i],Y_C6H6[i],color=colors[Y_C6H6[i]])

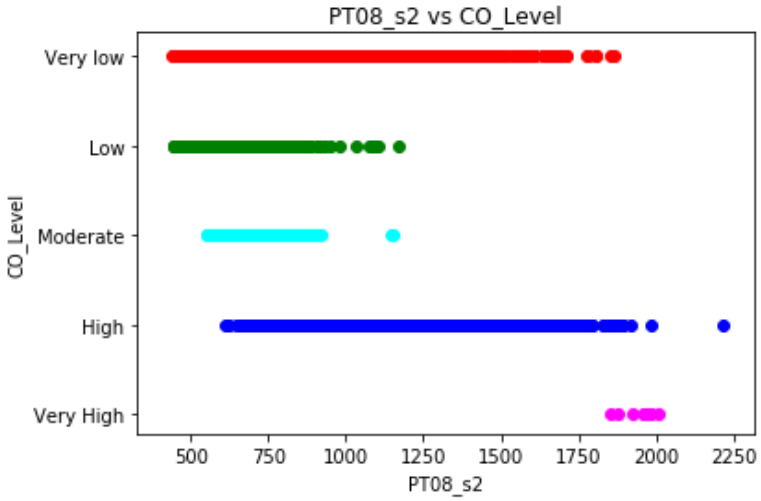
```
101.        plt.title('C6H6_GT vs CO_Level')
102.        plt.xlabel('C6H6_GT')
103.        plt.ylabel('CO_Level')
104.        plt.show()
```

O/P:



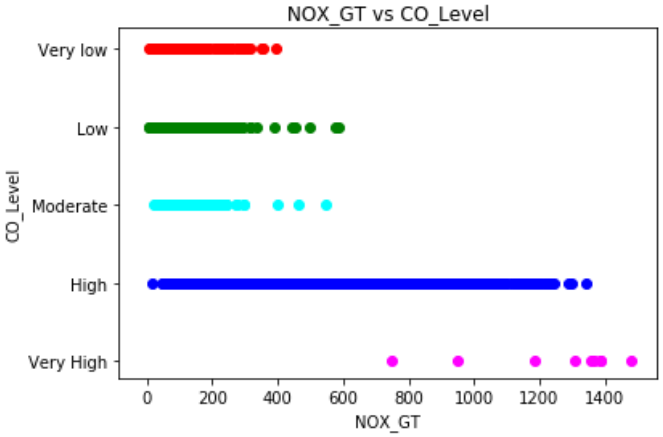

```
105.        #Plotting PT08_s2 vs CO_level

106.        #plt.xlim(1250,1500)
107.        for i in range(len(X_Pt08s2_)):
108.        plt.scatter(X_Pt08s2_[i],Y_Pt08s2[i],color=colors[Y_Pt08s2[i]])
109.        plt.title('PT08_s2 vs CO_Level')
110.        plt.xlabel('PT08_s2')
111.        plt.ylabel('CO_Level')
112.        plt.show()
```

O/P:



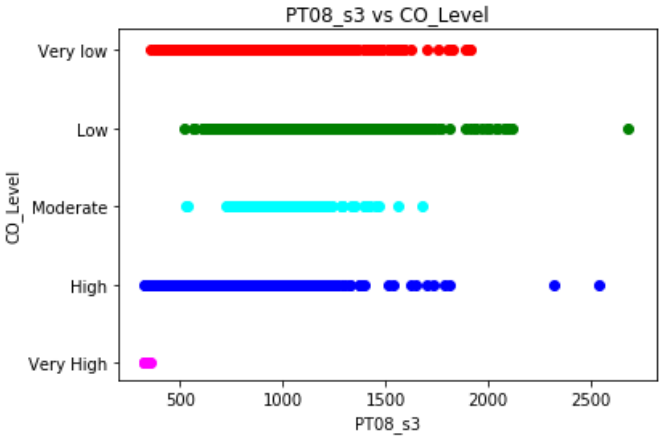
```
#Plotting NOX_GT vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_Nox_)):
plt.scatter(X_Nox_[i],Y_Nox[i],color=colors[Y_Nox[i]])
plt.title('NOX_GT vs CO_Level')
plt.xlabel('NOX_GT')
plt.ylabel('CO_Level')
plt.show()
```

O/P:



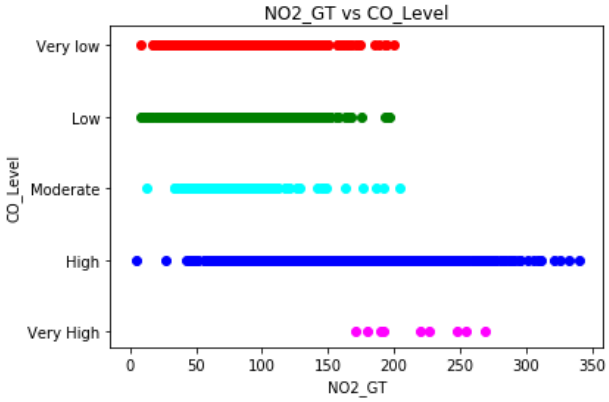

```
#Plotting PT08_s3 vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_Pt08s3_)):
plt.scatter(X_Pt08s3_[i],Y_Pt08s3[i],color=colors[Y_Pt08s3[i]])
plt.title('PT08_s3 vs CO_Level')
plt.xlabel('PT08_s3')
plt.ylabel('CO_Level')
plt.show()
```

O/P:



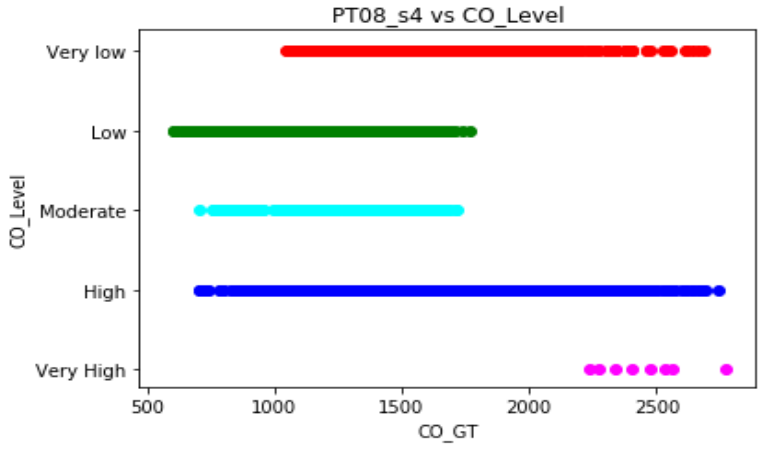
```
#Plotting NO2_GT vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_No2_)):
plt.scatter(X_No2_[i],Y_No2[i],color=colors[Y_No2[i]])
plt.title('NO2_GT vs CO_Level')
plt.xlabel('NO2_GT')
plt.ylabel('CO_Level')
plt.show()
```

O/P:



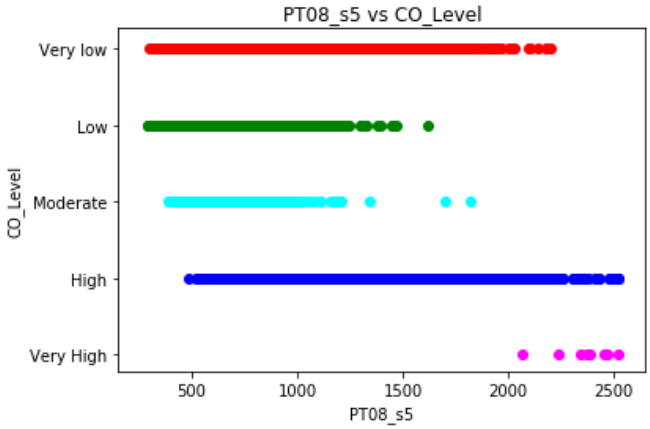

```
#Plotting PT08_s4 vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_Pt08s4_)):
plt.scatter(X_Pt08s4_[i],Y_Pt08s4[i],color=colors[Y_Pt08s4[i]])
plt.title('PT08_s4 vs CO_Level')
plt.xlabel('CO_GT')
plt.ylabel('CO_Level')
plt.show()
```

O/P:



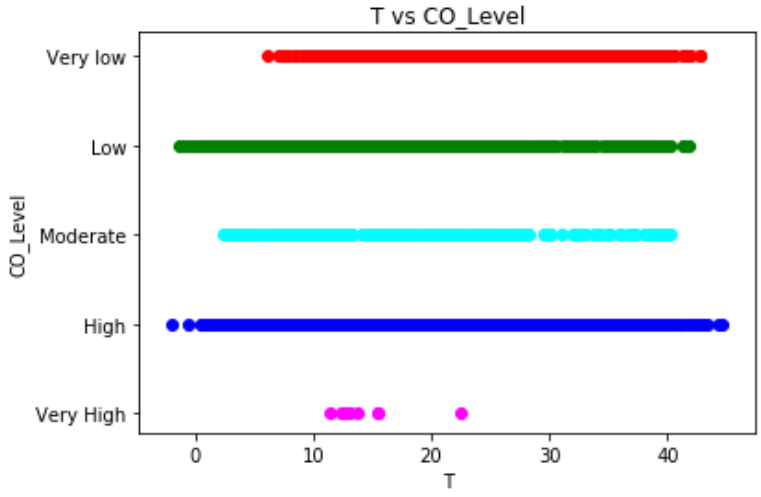
```
#Plotting PT08_s5 vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_Pt08s5_)):
plt.scatter(X_Pt08s5_[i],Y_Pt08s5[i],color=colors[Y_Pt08s5[i]])
plt.title('PT08_s5 vs CO_Level')
plt.xlabel('PT08_s5')
plt.ylabel('CO_Level')
plt.show()
```

O/P:



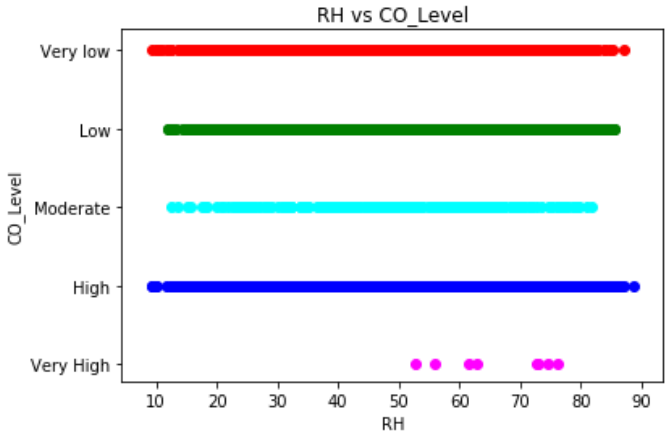

```
#Plotting T vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_T_)):
plt.scatter(X_T_[i],Y_T[i],color=colors[Y_T[i]])
plt.title('T vs CO_Level')
plt.xlabel('T')
plt.ylabel('CO_Level')
plt.show()
```

O/P:



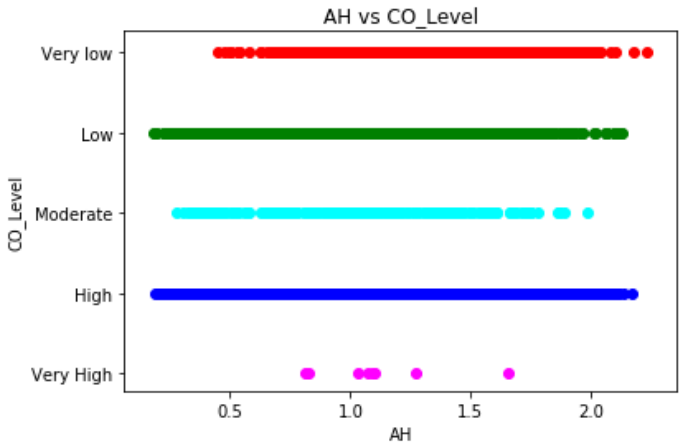
```
#Plotting RH vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_RH_)):
plt.scatter(X_RH_[i],Y_RH[i],color=colors[Y_RH[i]])
plt.title('RH vs CO_Level')
plt.xlabel('RH')
plt.ylabel('CO_Level')
plt.show()
```

O/P:




```
#Plotting AH vs CO_level

#plt.xlim(1250,1500)
for i in range(len(X_AH_)):
plt.scatter(X_AH_[i],Y_AH[i],color=colors[Y_AH[i]])
plt.title('AH vs CO_Level')
plt.xlabel('AH')
plt.ylabel('CO_Level')
plt.show()
```

O/P:



```
177.    #Plotting NOX_GT vs NO2_GT

178.    for i in range(len(X_Nox)):
179.    plt.scatter(X_Nox[i],X_No2[i])
180.    plt.title('NOX_GT vs NO2_GT')
181.    plt.xlabel('NOX_GT')
182.    plt.ylabel('NO2_GT')
183.    plt.grid(True)
184.    plt.show()
```

O/P: