

```
1. # -*- coding: utf-8 -*-
2. """
3. Created on Sat Mar 28 23:27:04 2020

4. @author: Admin
5. """
6. #Importing the libraries
7. import matplotlib.pyplot as plt
8. import numpy as np
9. import pandas as pd

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

12. X_COgt=dataset.iloc[:,2].values
13. X_Pt08s1=dataset.iloc[:,3].values
14. X_Nmhcg=dataset.iloc[:,4].values
15. X_C6H6=dataset.iloc[:,5].values
16. X_Pt08s2=dataset.iloc[:,6].values
17. X_No=dataset.iloc[:,7].values
18. X_Pt08s3=dataset.iloc[:,8].values
19. X_No2=dataset.iloc[:,9].values
20. X_Pt08s4=dataset.iloc[:,10].values
21. X_Pt08s5=dataset.iloc[:,11].values
22. X_T=dataset.iloc[:,12].values
23. X_RH=dataset.iloc[:,13].values
24. X_AH=dataset.iloc[:,14].values

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

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

30. *#Removing -200 from columns and accordingly change the CO\_level column*

31. `X_Cogt=X_COgt[X_COgt!=-200]`

32. `X_Pt08s1=X_Pt08s1[X_Pt08s1!=-200]`

33. `X_Nmhcgt=X_Nmhcgt[X_Nmhcgt!=-200]`

34. `X_C6H6=X_C6H6[X_C6H6!=-200]`

35. `X_Pt08s2=X_Pt08s2[X_Pt08s2!=-200]`

36. `X_NoX=X_NoX[X_NoX!=-200]`

37. `X_Pt08s3=X_Pt08s3[X_Pt08s3!=-200]`

38. `X_No2=X_No2[X_No2!=-200]`

39. `X_Pt08s4=X_Pt08s4[X_Pt08s4!=-200]`

40. `X_Pt08s5=X_Pt08s5[X_Pt08s5!=-200]`

41. `X_T=X_T[X_T!=-200]`

42. `X_RH=X_RH[X_RH!=-200]`

43. `X_AH=X_AH[X_AH!=-200]`

44. `a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13=[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]`

45. `for i in range(0,7485):`

46. `if X_COgt[i]==-200:a1.append(i)`

47. `if X_Pt08s1[i]==-200:a2.append(i)`

48. `if X_Nmhcgt[i]==-200:a3.append(i)`

49. `if X_C6H6[i]==-200:a4.append(i)`

50. `if X_Pt08s2[i]==-200:a5.append(i)`

51. `if X_NoX[i]==-200:a6.append(i)`

52. `if X_Pt08s3[i]==-200:a7.append(i)`

53. `if X_No2[i]==-200:a8.append(i)`

54. `if X_Pt08s4[i]==-200:a9.append(i)`

55. `if X_Pt08s5[i]==-200:a10.append(i)`

56. `if X_T[i]==-200:a11.append(i)`

57. `if X_RH[i]==-200:a12.append(i)`

58. `if X_AH[i]==-200:a13.append(i)`

59. `Y_Cogt=np.delete(Y,a1)`

60. `Y_Pt08s1=np.delete(Y,a2)`

61. `Y_Nmhcgt=np.delete(Y,a3)`

62. `Y_C6H6=np.delete(Y,a4)`

63. `Y_Pt08s2=np.delete(Y,a5)`

64. `Y_NoX=np.delete(Y,a6)`

65. `Y_Pt08s3=np.delete(Y,a7)`

66. `Y_No2=np.delete(Y,a8)`

67. `Y_Pt08s4=np.delete(Y,a9)`

68. `Y_Pt08s5=np.delete(Y,a10)`

69. `Y_T=np.delete(Y,a11)`

70. `Y_RH=np.delete(Y,a12)`

71. `Y_AH=np.delete(Y,a13)`

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

```
73. #Plotting CO_GT vs CO_level
```

```
74. #plt.xlim(1250,1500)
```

```
75. for i in range(len(X_Cogt_)):
```

```
76. plt.scatter(X_Cogt_[i],Y_Cogt[i],color=colors[Y_Cogt[i]])
```

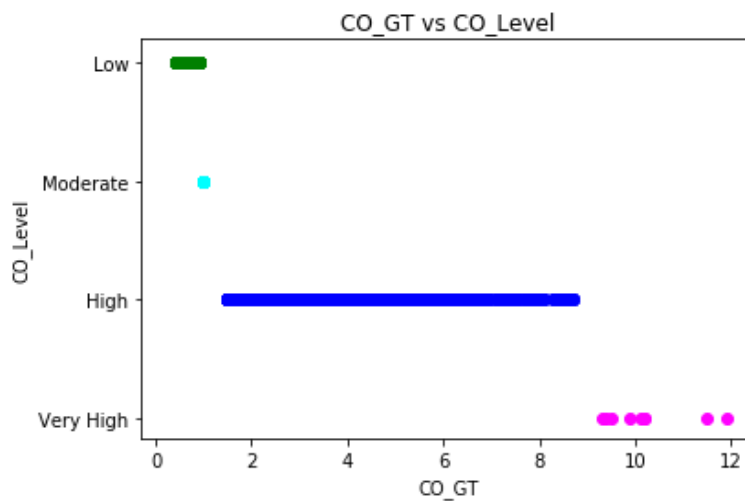
```
77. plt.title('CO_GT vs CO_Level')
```

```
78. plt.xlabel('CO_GT')
```

```
79. plt.ylabel('CO_Level')
```

```
80. plt.show()
```

O/P:



```
81. #Plotting PT08_S1 vs CO_level
```

```
82. #plt.xlim(1250,1500)
```

```
83. for i in range(len(X_Pt08s1_)):
```

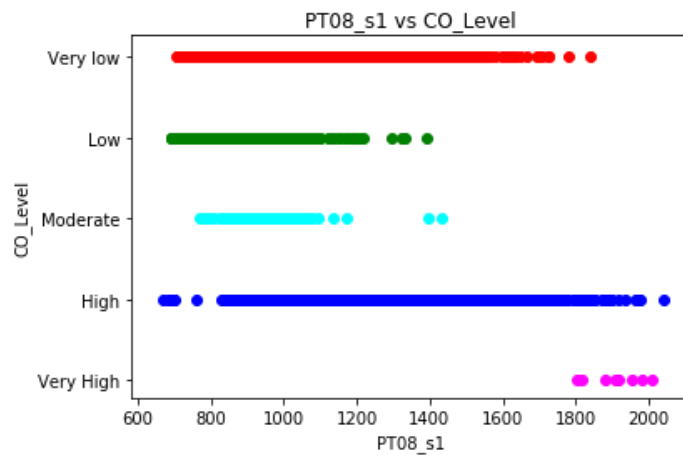
```
84. plt.scatter(X_Pt08s1_[i],Y_Pt08s1[i],color=colors[Y_Pt08s1[i]])
```

```
85. plt.title('PT08_s1 vs CO_Level')
```

```
86. plt.xlabel('PT08_s1')
```

```
87. plt.ylabel('CO_Level')
```

```
88. 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]])

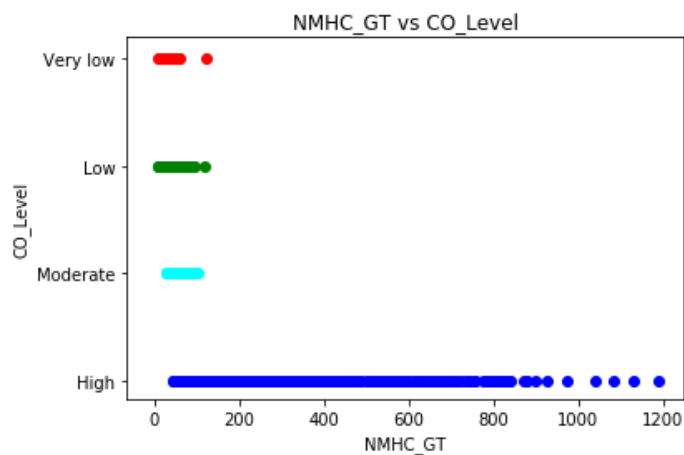
93. plt.title('NMHC\_GT vs CO\_Level')

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\_)):

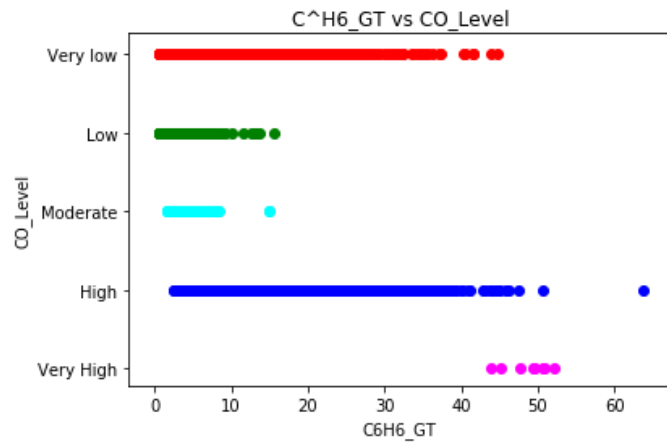
100. plt.scatter(X\_C6H6\_[i],Y\_C6H6[i],color=colors[Y\_C6H6[i]])

```

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

```

O/P:

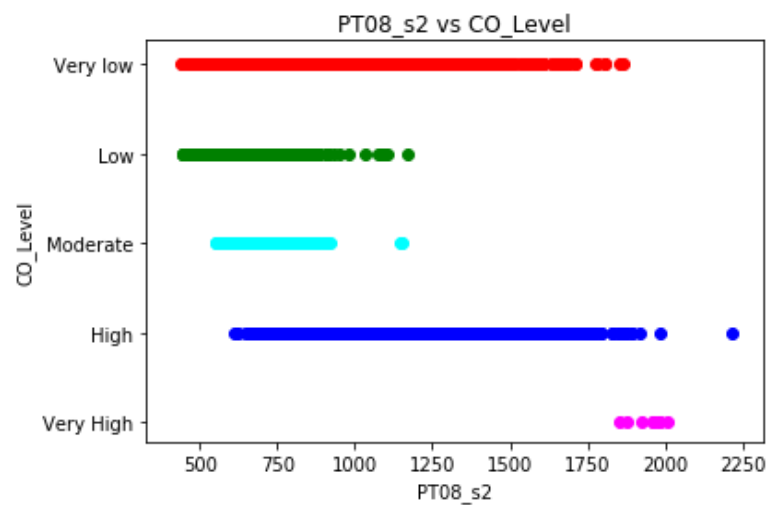


```

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:

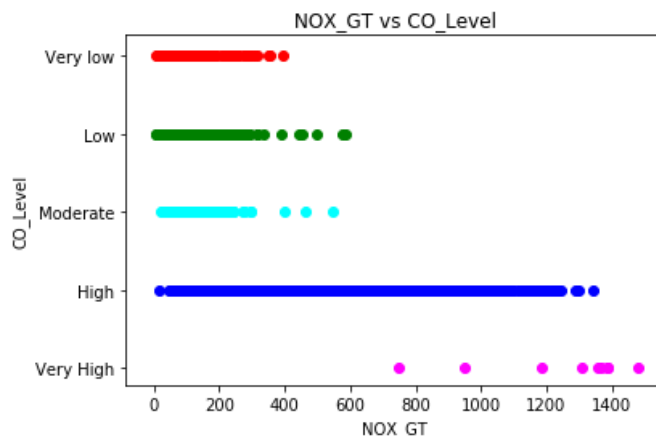


```

113.      #Plotting NOX_GT vs CO_level
114.      #plt.xlim(1250,1500)
115.      for i in range(len(X_NoX_)):
116.          plt.scatter(X_NoX_[i],Y_NoX[i],color=colors[Y_NoX[i]])
117.          plt.title('NOX_GT vs CO_Level')
118.          plt.xlabel('NOX_GT')
119.          plt.ylabel('CO_Level')
120.          plt.show()

```

O/P:

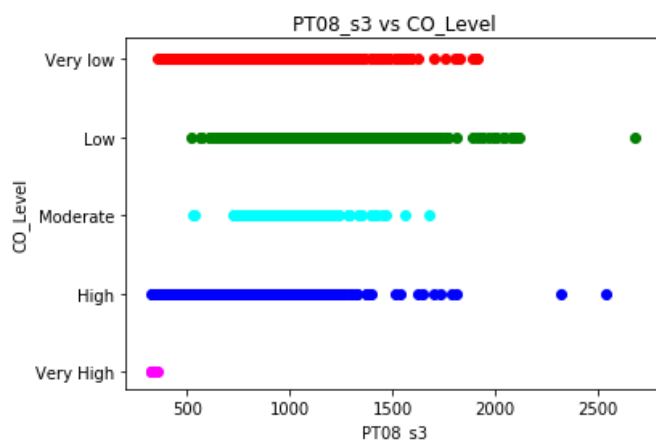


```

121.      #Plotting PT08_s3 vs CO_level
122.      #plt.xlim(1250,1500)
123.      for i in range(len(X_Pt08s3_)):
124.          plt.scatter(X_Pt08s3_[i],Y_Pt08s3[i],color=colors[Y_Pt08s3[i]])
125.          plt.title('PT08_s3 vs CO_Level')
126.          plt.xlabel('PT08_s3')
127.          plt.ylabel('CO_Level')
128.          plt.show()

```

O/P:

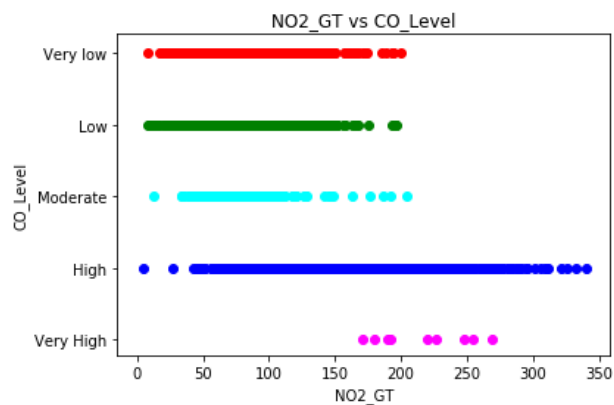


```

129.      #Plotting NO2_GT vs CO_level
130.      #plt.xlim(1250,1500)
131.      for i in range(len(X_No2_)):
132.          plt.scatter(X_No2_[i],Y_No2[i],color=colors[Y_No2[i]])
133.          plt.title('NO2_GT vs CO_Level')
134.          plt.xlabel('NO2_GT')
135.          plt.ylabel('CO_Level')
136.          plt.show()

```

O/P:

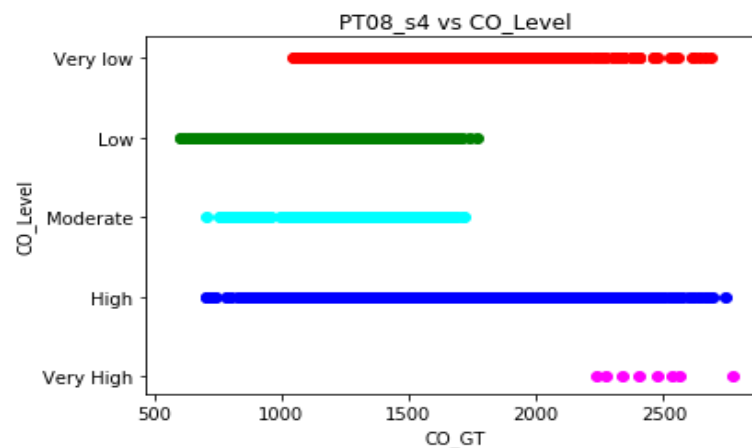


```

137.      #Plotting PT08_s4 vs CO_level
138.      #plt.xlim(1250,1500)
139.      for i in range(len(X_Pt08s4_)):
140.          plt.scatter(X_Pt08s4_[i],Y_Pt08s4[i],color=colors[Y_Pt08s4[i]])
141.          plt.title('PT08_s4 vs CO_Level')
142.          plt.xlabel('CO_GT')
143.          plt.ylabel('CO_Level')
144.          plt.show()

```

O/P:

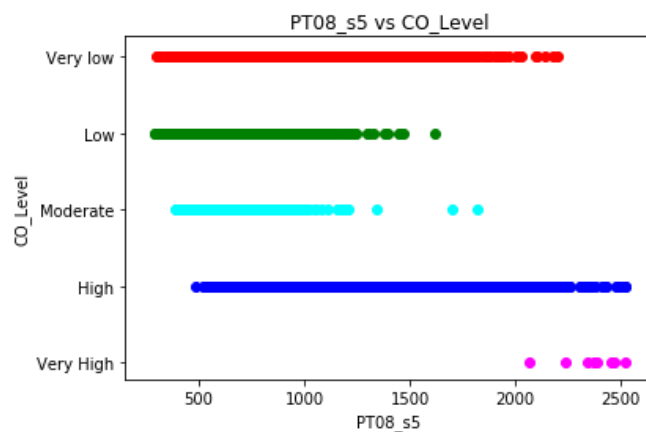


```

145.      #Plotting PT08_s5 vs CO_level
146.      #plt.xlim(1250,1500)
147.      for i in range(len(X_Pt08s5_)):
148.          plt.scatter(X_Pt08s5_[i],Y_Pt08s5[i],color=colors[Y_Pt08s5[i]])
149.          plt.title('PT08_s5 vs CO_Level')
150.          plt.xlabel('PT08_s5')
151.          plt.ylabel('CO_Level')
152.          plt.show()

```

O/P:

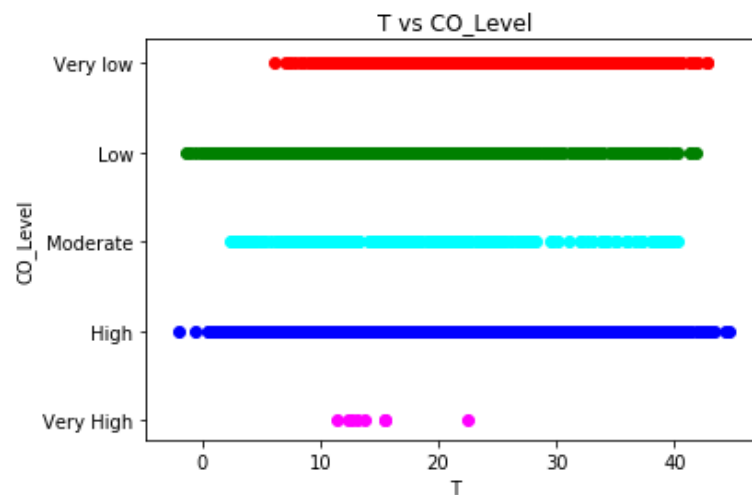


```

153.      #Plotting T vs CO_level
154.      #plt.xlim(1250,1500)
155.      for i in range(len(X_T_)):
156.          plt.scatter(X_T_[i],Y_T[i],color=colors[Y_T[i]])
157.          plt.title('T vs CO_Level')
158.          plt.xlabel('T')
159.          plt.ylabel('CO_Level')
160.          plt.show()

```

O/P:



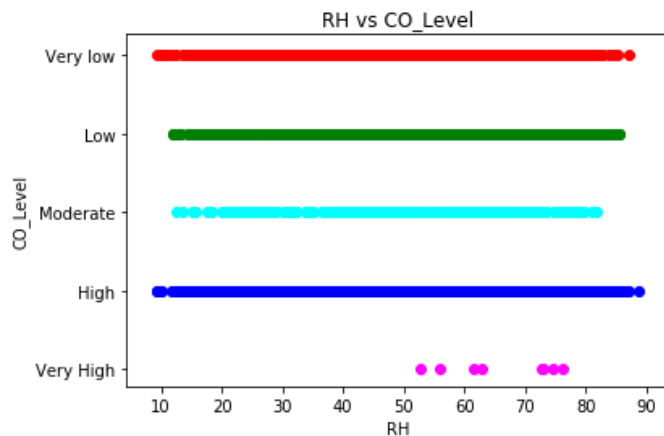


```

161.      #Plotting RH vs CO_level
162.      #plt.xlim(1250,1500)
163.      for i in range(len(X_RH_)):
164.          plt.scatter(X_RH_[i],Y_RH[i],color=colors[Y_RH[i]])
165.          plt.title('RH vs CO_Level')
166.          plt.xlabel('RH')
167.          plt.ylabel('CO_Level')
168.          plt.show()

```

O/P:

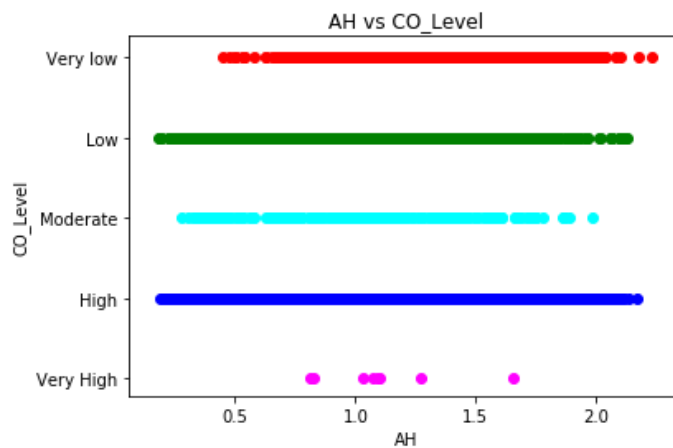


```

169.      #Plotting AH vs CO_level
170.      #plt.xlim(1250,1500)
171.      for i in range(len(X_AH_)):
172.          plt.scatter(X_AH_[i],Y_AH[i],color=colors[Y_AH[i]])
173.          plt.title('AH vs CO_Level')
174.          plt.xlabel('AH')
175.          plt.ylabel('CO_Level')
176.          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:

