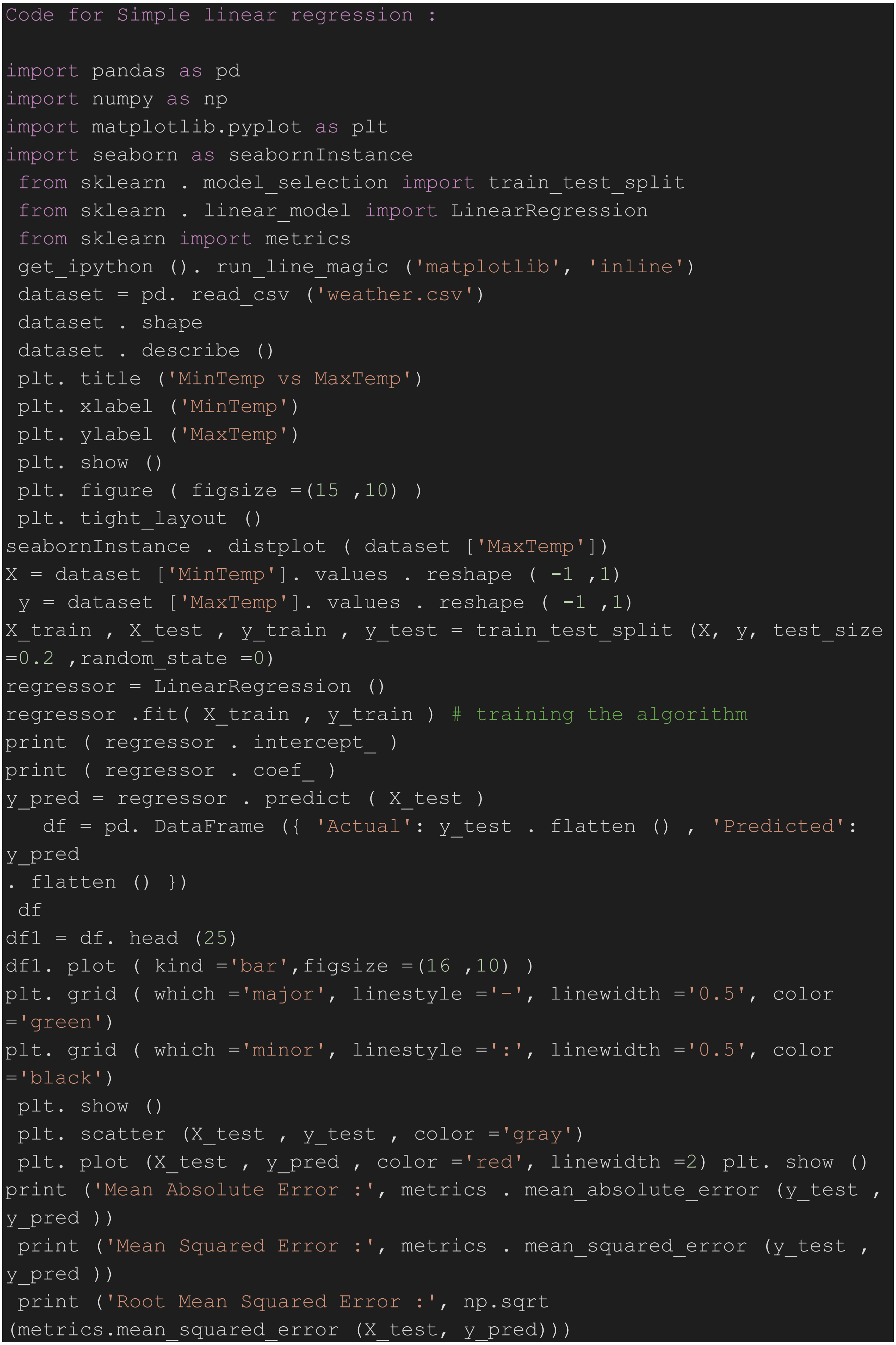
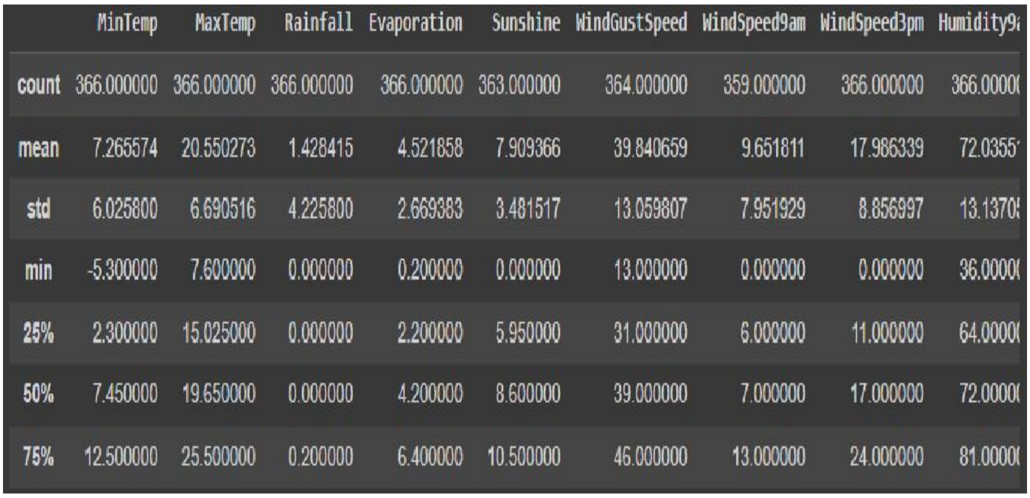


```
Code for Simple linear regression :

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as seabornInstance
 from sklearn . model_selection import train_test_split
 from sklearn . linear_model import LinearRegression
 from sklearn import metrics
 get_ipython (). run_line_magic ('matplotlib', 'inline')
 dataset = pd. read_csv ('weather.csv')
 dataset . shape
 dataset . describe ()
 plt. title ('MinTemp vs MaxTemp')
 plt. xlabel ('MinTemp')
 plt. ylabel ('MaxTemp')
 plt. show ()
 plt. figure ( figsize =(15 ,10) )
 plt. tight_layout ()
seabornInstance . distplot ( dataset ['MaxTemp'])
X = dataset ['MinTemp']. values . reshape ( -1 ,1)
 y = dataset ['MaxTemp']. values . reshape ( -1 ,1)
X_train , X_test , y_train , y_test = train_test_split (X, y, test_size
=0.2 ,random_state =0)
regressor = LinearRegression ()
regressor .fit( X_train , y_train ) # training the algorithm
print ( regressor . intercept_ )
print ( regressor . coef_ )
y_pred = regressor . predict ( X_test )
   df = pd. DataFrame ({ 'Actual': y_test . flatten () , 'Predicted':
y_pred
. flatten () })
 df
df1 = df. head (25)
df1. plot ( kind ='bar',figsize =(16 ,10) )
plt. grid ( which ='major', linestyle ='-', linewidth ='0.5', color
='green')
plt. grid ( which ='minor', linestyle =':', linewidth ='0.5', color
='black')
 plt. show ()
 plt. scatter (X_test , y_test , color ='gray')
 plt. plot (X_test , y_pred , color ='red', linewidth =2) plt. show ()
print ('Mean Absolute Error :', metrics . mean_absolute_error (y_test ,
y_pred ))
 print ('Mean Squared Error :', metrics . mean_squared_error (y_test ,
y_pred ))
 print ('Root Mean Squared Error :', np.sqrt
(metrics.mean_squared_error (X_test, y_pred)))
```

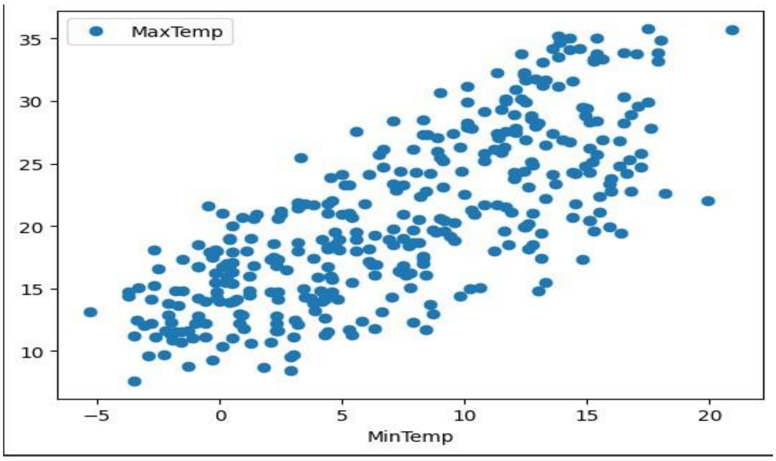
Output :

Dataset.describe():

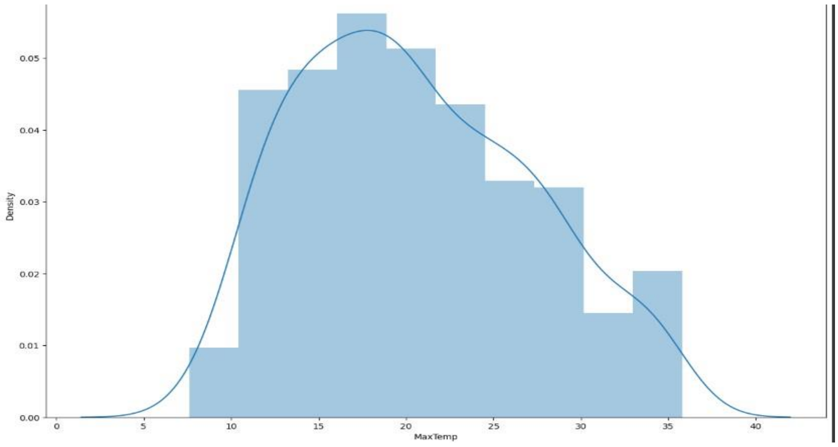


	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9a
count	366.000000	366.000000	366.000000	366.000000	363.000000	364.000000	359.000000	366.000000	366.00000
mean	7.265574	20.550273	1.428415	4.521858	7.909366	39.840659	9.651811	17.986339	72.0355
std	6.025800	6.690516	4.225800	2.669383	3.481517	13.059807	7.951929	8.856997	13.1370
min	-5.300000	7.600000	0.000000	0.200000	0.000000	13.000000	0.000000	0.000000	36.0000
25%	2.300000	15.025000	0.000000	2.200000	5.950000	31.000000	6.000000	11.000000	64.0000
50%	7.450000	19.650000	0.000000	4.200000	8.600000	39.000000	7.000000	17.000000	72.0000
75%	12.500000	25.500000	0.200000	6.400000	10.500000	46.000000	13.000000	24.000000	81.0000

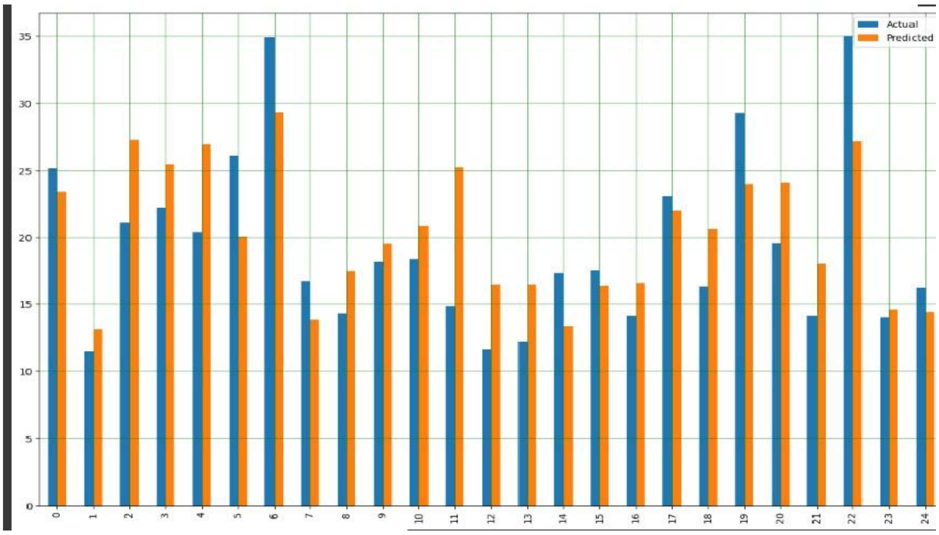
plt.show():



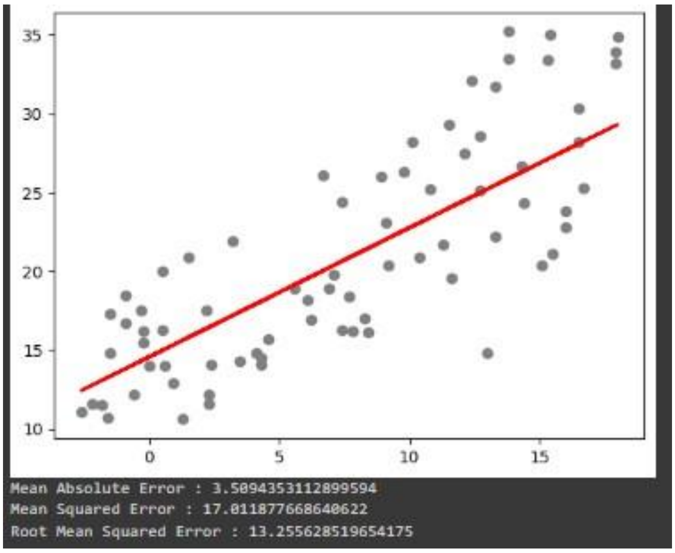
Plotting of MaxTemp :



Plotting graph for actual and predicted data :

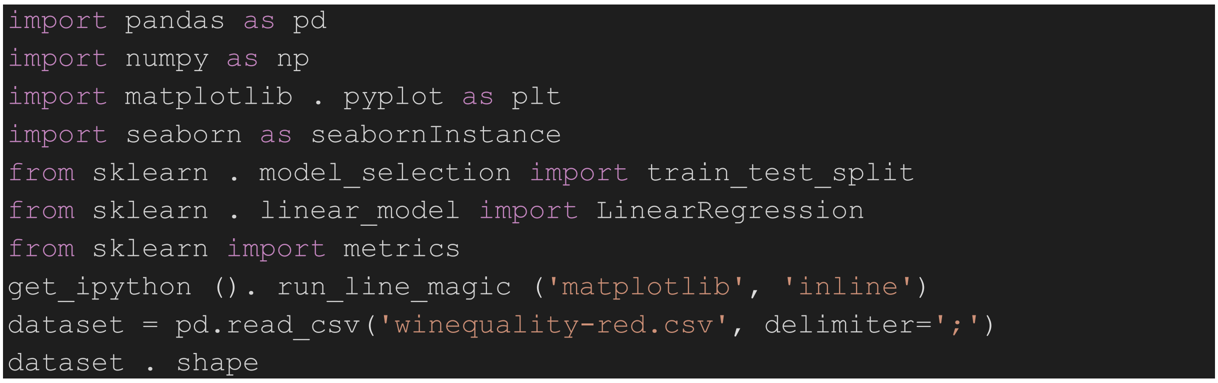


Line plotted for simple regression :



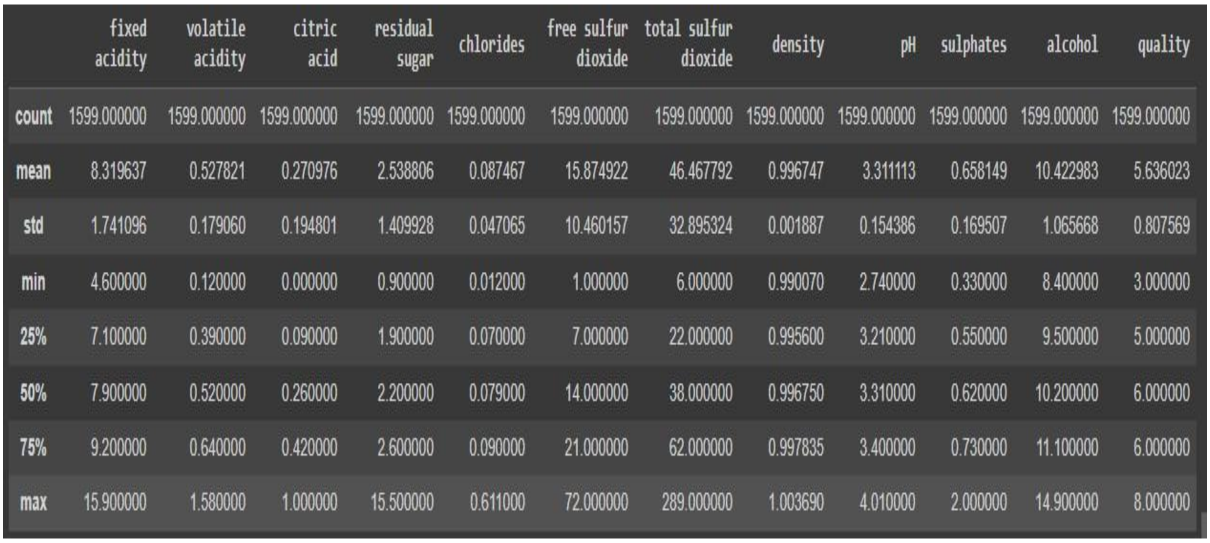
```
Mean Absolute Error : 3.5094353112899594
Mean Squared Error : 17.011877668640622
Root Mean Squared Error : 13.255628519654175
```

Code for Multiple Linear regression :



```
import pandas as pd
import numpy as np
import matplotlib . pyplot as plt
import seaborn as seabornInstance
from sklearn . model_selection import train_test_split
from sklearn . linear_model import LinearRegression
from sklearn import metrics
get_ipython (). run_line_magic ('matplotlib', 'inline')
dataset = pd.read_csv('winequality-red.csv', delimiter=';')
dataset . shape
dataset . describe () dataset.describe () dataset . isnull () .any ()
dataset = dataset . fillna ( method ='ffill')
X = dataset[['fixed acidity', 'volatile acidity', 'citric acid',
'residual
sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide',
'density', 'pH', 'sulphates', 'alcohol']].values
y = dataset['quality'].values
plt . figure ( figsize =(15 ,10) ) plt . tight_layout ()
seabornInstance . distplot ( dataset ['quality'])
X_train , X_test , y_train , y_test = train_test_split (X , y ,
test_size =0.2 , random_state =0) regressor = LinearRegression ()
regressor . fit ( X_train , y_train )
coeff_df = pd . DataFrame ( regressor . coef_ , dataset . columns
[0:11] ,
columns =['Coefficient'])coeff_df y_pred = regressor . predict ( X_test
) df = pd . DataFrame ({ 'Actual': y_test , 'Predicted': y_pred })
 df1 = df . head (25) df1 df1 . plot ( kind ='bar', figsize =(10 ,8) )
 plt . grid ( which ='major', linestyle ='-', linewidth ='0.5', color
='green')
 plt . grid ( which ='minor', linestyle =':', linewidth ='0.5', color
='black') plt . show ()
print ('Mean Absolute Error :', metrics . mean_absolute_error ( y_test,
y_pred ) )
print ('Mean Squared Error :', metrics . mean_squared_error ( y_test ,
y_pred ) ) print ('Root Mean Squared Error :', np . sqrt ( metrics .
mean_squared_error ( y_test ,y_pred ) ) )
```

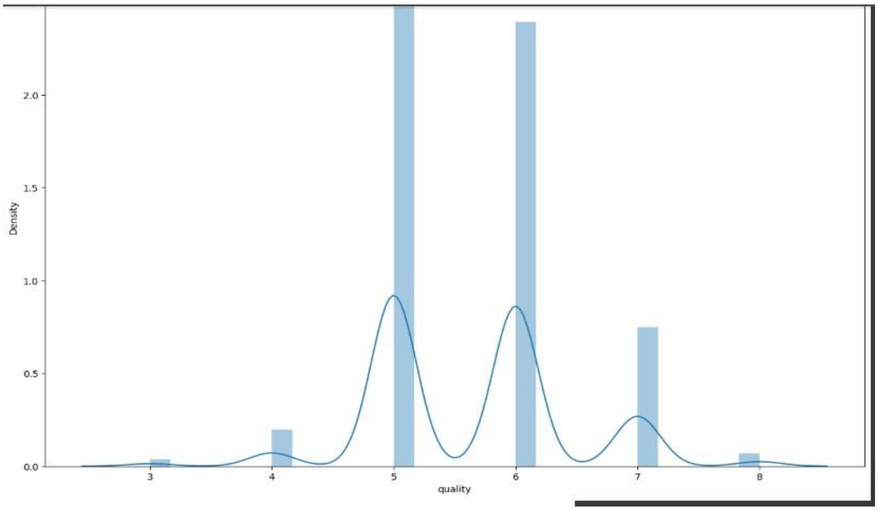
dataset.describe() :



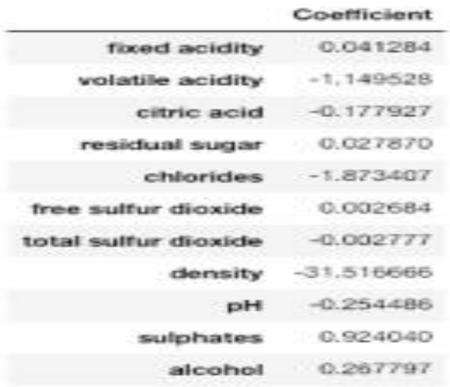
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

Graph for quality

This function is used to plot datapoints between min Temp and Max attributes.

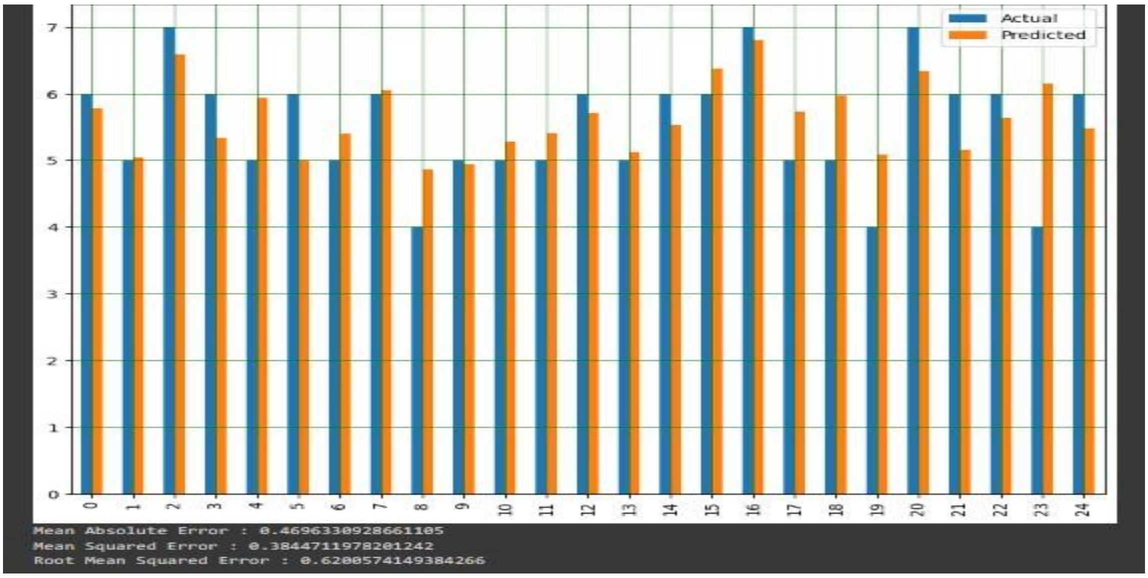


List of coefficients for different attributes :



	Coefficient
fixed acidity	0.041284
volatile acidity	-1,149528
citric acid	-0.177927
residual sugar	0.027870
chlorides	-1.873407
free sulfur dioxide	0.002684
total sulfur dioxide	-0.002777
density	-31.516666
pH	-0.254486
sulphates	0.924040
alcohol	0.267797

Plotting graph for actual and predicted data :



```
Mean Absolute Error : 0.4696330928661105
Mean Squared Error : 0.3844711978201242
Root Mean Squared Error : 0.6200574149384266
```