Importing Libraries:

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
```

Reading the dataset:

In [2]:

```
data=pd.read_csv('water_dataX.csv',encoding='ISO-8859-1',low_memory=False)
```

In [3]:

```
data.head()
```

Out[3]:

	STATION CODE	LOCATIONS	STATE	Temp	D.O. (mg/l)	PH	CONDUCTIVITY (µmhos/cm)	B.O.D. (mg/l)	NITRATENAN N+ NITRITENANN (mg/l)
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	NAN	0.1
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	2	0.2
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	1.7	0.1
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	3.8	0.5
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	1.9	0.4

Analysing the data:

In [4]:

```
data.describe()
```

Out[4]:

	year
count	1991.000000
mean	2010.038172
std	3.057333
min	2003.000000
25%	2008.000000
50%	2011.000000
75%	2013.000000
max	2014.000000

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1991 entries, 0 to 1990
Data columns (total 12 columns):
 #   Column                              Non-Null Count  Dtype 
---  ------                              --------------  ----- 
 0   STATION CODE                        1991 non-null   object
 1   LOCATIONS                           1991 non-null   object
 2   STATE                               1991 non-null   object
 3   Temp                                1991 non-null   object
 4   D.O. (mg/l)                         1991 non-null   object
 5   PH                                  1991 non-null   object
 6   CONDUCTIVITY (µmhos/cm)             1991 non-null   object
 7   B.O.D. (mg/l)                       1991 non-null   object
 8   NITRATENAN N+ NITRITENANN (mg/l)    1991 non-null   object
 9   FECAL COLIFORM (MPN/100ml)          1991 non-null   object
 10  TOTAL COLIFORM (MPN/100ml)Mean      1991 non-null   object
 11  year                                1991 non-null   int64 
dtypes: int64(1), object(11)
memory usage: 186.8+ KB
```

In [6]:

```
data.shape
```

Out[6]:

```
(1991, 12)
```

Handling Missing Values:

In [7]:

```
data.isnull().any()
```

Out[7]:

```
STATION CODE                         False
LOCATIONS                            False
STATE                                False
Temp                                 False
D.O. (mg/l)                          False
PH                                   False
CONDUCTIVITY (µmhos/cm)              False
B.O.D. (mg/l)                        False
NITRATENAN N+ NITRITENANN (mg/l)     False
FECAL COLIFORM (MPN/100ml)           False
TOTAL COLIFORM (MPN/100ml)Mean       False
year                                 False
dtype: bool
```

In [8]:

```
data.isnull().sum()
```

Out[8]:

```
STATION CODE                         0
LOCATIONS                            0
STATE                                0
Temp                                 0
D.O. (mg/l)                          0
PH                                   0
CONDUCTIVITY (µmhos/cm)              0
B.O.D. (mg/l)                        0
NITRATENAN N+ NITRITENANN (mg/l)     0
FECAL COLIFORM (MPN/100ml)           0
TOTAL COLIFORM (MPN/100ml)Mean       0
year                                 0
dtype: int64
```

In [9]:

```
data.dtypes
```

Out[9]:

```
STATION CODE                         object
LOCATIONS                            object
STATE                                object
Temp                                 object
D.O. (mg/l)                          object
PH                                   object
CONDUCTIVITY (µmhos/cm)              object
B.O.D. (mg/l)                        object
NITRATENAN N+ NITRITENANN (mg/l)     object
FECAL COLIFORM (MPN/100ml)           object
TOTAL COLIFORM (MPN/100ml)Mean       object
year                                  int64
dtype: object
```

In [10]:

```
data['Temp']=pd.to_numeric(data['Temp'],errors='coerce')
data['D.O. (mg/l)']=pd.to_numeric(data['D.O. (mg/l)'],errors='coerce')
data['PH']=pd.to_numeric(data['PH'],errors='coerce')
data['CONDUCTIVITY (µmhos/cm)']=pd.to_numeric(data['CONDUCTIVITY (µmhos/cm)'],errors='coerc
data['B.O.D. (mg/l)']=pd.to_numeric(data['B.O.D. (mg/l)'],errors='coerce')
data['NITRATENAN N+ NITRITENANN (mg/l)']=pd.to_numeric(data['NITRATENAN N+ NITRITENANN (mg/
data['TOTAL COLIFORM (MPN/100ml)Mean']=pd.to_numeric(data['TOTAL COLIFORM (MPN/100ml)Mean']
data.dtypes
```

Out[10]:

```
STATION CODE                          object
LOCATIONS                             object
STATE                                 object
Temp                                 float64
D.O. (mg/l)                          float64
PH                                   float64
CONDUCTIVITY (µmhos/cm)              float64
B.O.D. (mg/l)                        float64
NITRATENAN N+ NITRITENANN (mg/l)     float64
FECAL COLIFORM (MPN/100ml)            object
TOTAL COLIFORM (MPN/100ml)Mean       float64
year                                   int64
dtype: object
```

In [11]:

```
data.isnull().sum()
```

Out[11]:

```
STATION CODE                          0
LOCATIONS                             0
STATE                                 0
Temp                                 92
D.O. (mg/l)                          31
PH                                    8
CONDUCTIVITY (µmhos/cm)              25
B.O.D. (mg/l)                        43
NITRATENAN N+ NITRITENANN (mg/l)    225
FECAL COLIFORM (MPN/100ml)            0
TOTAL COLIFORM (MPN/100ml)Mean      132
year                                  0
dtype: int64
```

In [12]:

```
data['Temp'].fillna(data['Temp'].mean(),inplace=True)
data['D.O. (mg/l)'].fillna(data['D.O. (mg/l)'].mean(),inplace=True)
data['PH'].fillna(data['PH'].mean(),inplace=True)
data['CONDUCTIVITY (µmhos/cm)'].fillna(data['CONDUCTIVITY (µmhos/cm)'].mean(),inplace=True)
data['B.O.D. (mg/l)'].fillna(data['B.O.D. (mg/l)'].mean(),inplace=True)
data['NITRATENAN N+ NITRITENANN (mg/l)'].fillna(data['NITRATENAN N+ NITRITENANN (mg/l)'].me
data['TOTAL COLIFORM (MPN/100ml)Mean'].fillna(data['TOTAL COLIFORM (MPN/100ml)Mean'].mean()
```

In [13]:

```
data.isnull().any()
```

Out[13]:

```
STATION CODE                          False
LOCATIONS                             False
STATE                                 False
Temp                                  False
D.O. (mg/l)                           False
PH                                    False
CONDUCTIVITY (µmhos/cm)               False
B.O.D. (mg/l)                         False
NITRATENAN N+ NITRITENANN (mg/l)      False
FECAL COLIFORM (MPN/100ml)            False
TOTAL COLIFORM (MPN/100ml)Mean        False
year                                  False
dtype: bool
```

In [14]:

```
data.drop("FECAL COLIFORM (MPN/100ml)",axis=1,inplace=True)
```

In [15]:

```
data.head()
```

Out[15]:

	STATION CODE	LOCATIONS	STATE	Temp	D.O. (mg/l)	PH	CONDUCTIVITY (µmhos/cm)	B.O.D. (mg/l)	NITRATENAN NITRITENAN (mg
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203.0	6.940049	0
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189.0	2.000000	0
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179.0	1.700000	0
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64.0	3.800000	0
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83.0	1.900000	0

In [16]:

```
data=data.rename(columns={'D.O. (mg/l)':'do'})
data=data.rename(columns={'CONDUCTIVITY (µmhos/cm)':'co'})
data=data.rename(columns={'B.O.D. (mg/l)':'bod'})
data=data.rename(columns={'NITRATENAN N+ NITRITENANN (mg/l)':'na'})
data=data.rename(columns={'TOTAL COLIFORM (MPN/100ml)Mean':'tc'})
data=data.rename(columns={'STATION CODE':'station'})
data=data.rename(columns={'LOCATIONS':'location'})
data=data.rename(columns={'STATE':'state'})
data=data.rename(columns={'PH':'ph'})
```

In [17]:

```
data.head()
```

Out[17]:

	station	location	state	Temp	do	ph	co	bod	na	tc	year
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203.0	6.940049	0.1	27.0	2014
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189.0	2.000000	0.2	8391.0	2014
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179.0	1.700000	0.1	5330.0	2014
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64.0	3.800000	0.5	8443.0	2014
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83.0	1.900000	0.4	5500.0	2014

Water Quality Index Calculation:

In [18]:

```
data['npH']=data.ph.apply(lambda x:(100 if(8.5>=x>=7)
                                    else(80 if(8.6>=x>=8.5) or (6.9>=x>=6.8)
                                        else(60 if(8.8>=x>=8.6) or (6.8>=x>=6.7)
                                            else(40 if(9>=x>=8.8) or (6.7>=x>=6.5)
                                                else 0)))))
```

In [19]:

```
data['ndo']=data.do.apply(lambda x:(100 if(x>=6)
                                    else(80 if(6>=x>=5.1)
                                        else(60 if(5>=x>=4.1)
                                            else(40 if(4>=x>=3)
                                                else 0)))))
```

In [20]:

```
data['nco']=data.tc.apply(lambda x:(100 if(5>=x>=0)
                                    else(80 if(50>=x>=5)
                                        else(60 if(500>=x>=50)
                                            else(40 if(10000>=x>=500)
                                                else 0)))))
```

In [21]:

```
data['nbdo']=data.bod.apply(lambda x:(100 if(3>=x>=0)
                                    else(80 if(6>=x>=3)
                                        else(60 if(80>=x>=6)
                                            else(40 if(125>=x>=80)
                                                else 0)))))
```

In [22]:

```
data['nec']=data.co.apply(lambda x:(100 if(75>=x>=0)
                                    else(80 if(150>=x>=75)
                                        else(60 if(225>=x>=150)
                                            else(40 if(300>=x>=225)
                                                else 0)))))
```

In [23]:

```
data['nna']=data.na.apply(lambda x:(100 if(20>=x>=0)
                                    else(80 if(50>=x>=20)
                                        else(60 if(100>=x>=50)
                                            else(40 if(200>=x>=100)
                                                else 0)))))
```

In [24]:

```
data['wph']=data.npH*0.165
data['wdo']=data.ndo*0.281
data['wbdo']=data.nbdo*0.234
data['wec']=data.nec*0.009
data['wna']=data.nna*0.028
data['wco']=data.nco*0.281
data['wqi']=data.wph+data.wdo+data.wbdo+data.wec+data.wna+data.wco
data
```

Out[24]:

	station	location	state	Temp	do	ph	co	bod	na	
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.600000	6.7	7.5	203.0	6.940049	0.100000	
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.800000	5.7	7.2	189.0	2.000000	0.200000	83
2	1475	ZUARI AT PANCHAWADI	GOA	29.500000	6.3	6.9	179.0	1.700000	0.100000	53
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.700000	5.8	6.9	64.0	3.800000	0.500000	84
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.500000	5.8	7.3	83.0	1.900000	0.400000	55
...	...	...	...	...	...	...	...	...	...	
1986	1330	TAMBIRAPARANI AT ARUMUGANERI, TAMILNADU	NAN	26.209814	7.9	738.0	7.2	2.700000	0.518000	2
1987	1450	PALAR AT VANIYAMBADI WATER SUPPLY HEAD WORK, T...	NAN	29.000000	7.5	585.0	6.3	2.600000	0.155000	3
1988	1403	GUMTI AT U/S SOUTH TRIPURA,TRIPURA	NAN	28.000000	7.6	98.0	6.2	1.200000	1.623079	5
1989	1404	GUMTI AT D/S SOUTH TRIPURA, TRIPURA	NAN	28.000000	7.7	91.0	6.5	1.300000	1.623079	5
1990	1726	CHANDRAPUR, AGARTALA D/S OF HAORA RIVER, TRIPURA	NAN	29.000000	7.6	110.0	5.7	1.100000	1.623079	5

1991 rows × 24 columns

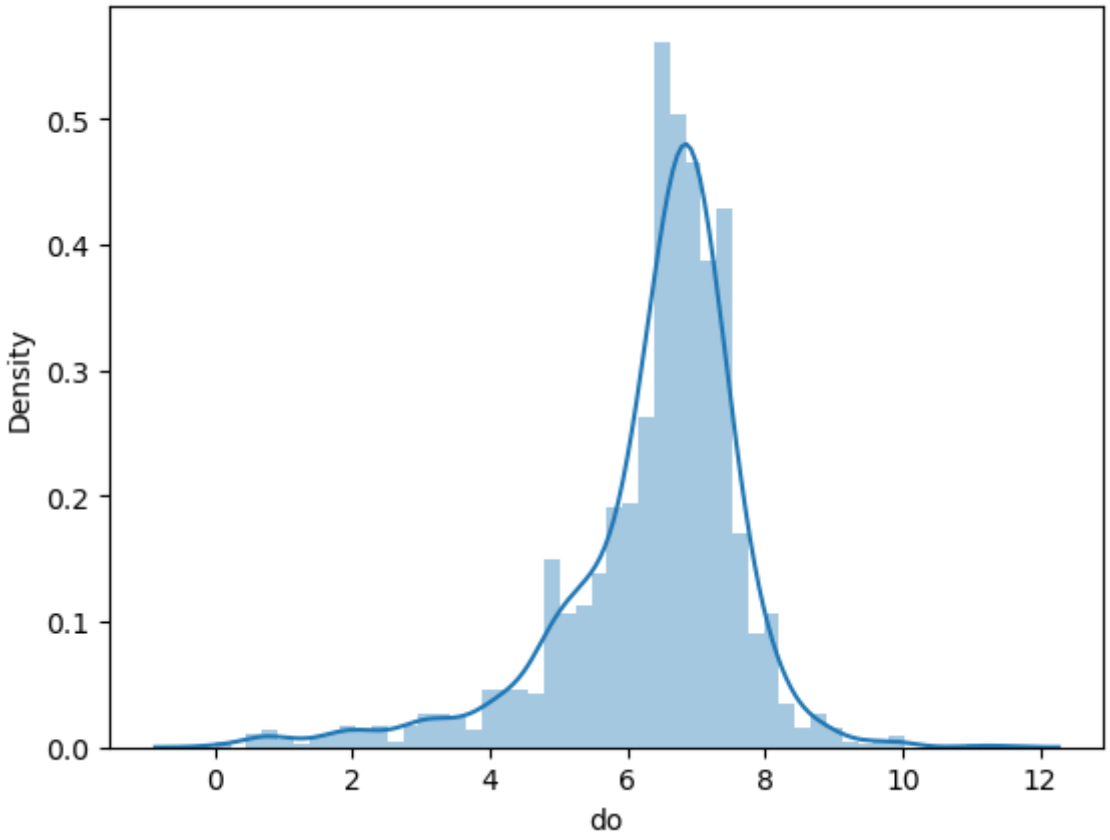
In [ ]:

Data Visualization:

In [25]:

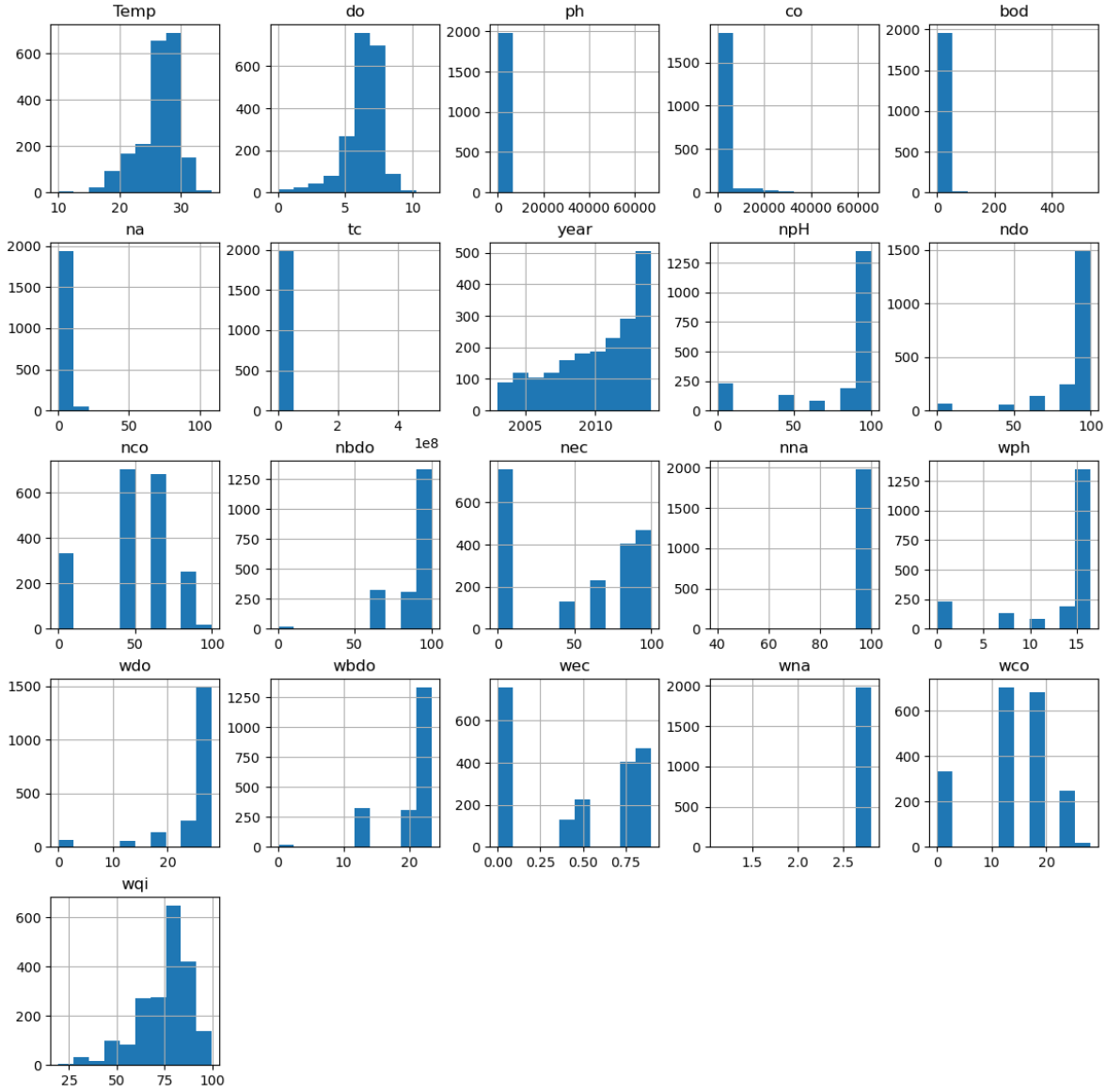
```
sns.distplot(data['do'])
plt.show()
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\distributions.py:2619: F
utureWarning: `distplot` is a deprecated function and will be removed in a f
uture version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function f
or histograms).
  warnings.warn(msg, FutureWarning)
```



In [26]:

```
data.hist(figsize=(14,14))
plt.show()
```



In [27]:

```
average=data.groupby('year')['wqi'].mean()
```

In [28]:

```
average.head()
```

Out[28]:

```
year
2003    66.239545
2004    61.290000
2005    73.762689
2006    72.909714
2007    74.233000
Name: wqi, dtype: float64
```

Model Building:

In [29]:

```
data.shape
```

Out[29]:

```
(1991, 24)
```

In [30]:

```
X=data.iloc[:,4:10].values
y=data.iloc[:,-1:].values
```

In [31]:

```
X.shape
```

Out[31]:

```
(1991, 6)
```

In [32]:

```
y.shape
```

Out[32]:

```
(1991, 1)
```

In [33]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=10)
```

In [34]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [35]:

```
from sklearn.ensemble import RandomForestRegressor
rfr=RandomForestRegressor()
rfr.fit(X_train,y_train)
y_pred=rfr.predict(X_test)
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_3744\1197134359.py:3: DataConve
rsionWarning: A column-vector y was passed when a 1d array was expected. Ple
ase change the shape of y to (n_samples,), for example using ravel().
  rfr.fit(X_train,y_train)
```

In [36]:

```
y_pred
```

Out[36]:

```
array([32.2264, 76.0866, 93.8218, 88.1964, 67.2204, 93.8798, 70.5976,
       82.742 , 30.4768, 88.1306, 88.249 , 82.94  , 71.8818, 79.64  ,
       82.94  , 81.7694, 33.8914, 72.4338, 69.946 , 73.1578, 77.5098,
       61.7072, 81.8912, 82.58  , 85.2564, 33.2098, 83.6478, 61.4344,
       72.4024, 88.2   , 66.1026, 74.858 , 88.38  , 50.1876, 85.2546,
       72.1804, 87.66  , 87.6636, 72.784 , 84.1568, 82.04  , 80.0142,
       72.06  , 45.1946, 88.56  , 71.3436, 68.3762, 92.7882, 66.4364,
       41.2778, 68.3034, 91.6598, 55.8088, 82.94  , 76.4192, 83.4736,
       82.04  , 71.4858, 78.6492, 70.8   , 83.5236, 61.08  , 88.7124,
       44.2458, 55.82  , 87.7086, 79.5034, 71.3118, 66.4962, 82.832 ,
       92.7126, 93.5926, 83.52  , 87.66  , 72.86  , 88.515 , 82.94  ,
       89.138 , 82.58  , 82.7564, 83.7468, 76.0444, 77.7356, 84.2362,
       88.4988, 65.4338, 62.2796, 83.7   , 94.18  , 70.9198, 76.223 ,
       77.3496, 66.5524, 84.8248, 55.9324, 43.4426, 82.76  , 43.4426,
       79.46  , 82.9998, 85.1234, 77.116 , 66.7878, 73.2086, 82.94  ,
       44.3582, 82.04  , 61.3838, 87.66  , 79.6596, 94.18  , 82.94  ,
       84.095 , 76.4346, 66.44  , 78.5952, 88.2   , 81.334 , 78.0746,
       77.0248, 89.046 , 82.0774, 83.7   , 72.06  , 73.0246, 71.8686,
       66.8274, 83.858 , 82.4   , 66.844 , 66.7538, 55.865 , 70.8108,
       66.44  , 87.7762, 72.06  , 82.94  , 92.7982, 69.4306, 93.3774,
       77.7426, 79.5022, 55.82  , 82.4   , 73.6336, 52.068 , 93.6184,
       91.0928, 78.857 , 82.58  , 88.38  , 82.94  , 88.038 , 76.1636,
       50.1492, 82.04  , 69.523 , 88.38  , 93.3774, 79.64  , 82.94  ,
       62.0138, 79.64  , 66.44  , 67.06  , 61.102 , 82.76  , 82.76  ,
       79.6364, 79.64  , 83.8012, 82.58  , 83.5848, 88.3818, 71.4906,
       79.5248, 88.56  , 88.2018, 82.94  , 82.4   , 78.2286, 69.5504,
       87.644 , 83.7   , 93.5996, 64.2034, 64.8282, 48.6416, 77.8064,
       50.2   , 73.0382, 83.2662, 81.5776, 50.2   , 82.58  , 94.122 ,
       88.3782, 71.2176, 82.049 , 56.6616, 70.8   , 62.527 , 82.98  ,
       76.0066, 60.9866, 72.4076, 87.66  , 82.76  , 88.3748, 87.987 ,
       88.873 , 93.28  , 73.8564, 72.0582, 82.5234, 88.8356, 71.4186,
       66.6198, 78.489 , 55.8008, 87.66  , 72.1624, 83.5236, 44.182 ,
       60.636 , 72.5956, 80.7596, 72.2678, 82.04  , 82.76  , 78.439 ,
       89.2912, 79.5032, 82.76  , 82.76  , 87.5994, 94.0036, 81.3766,
       44.4656, 67.3638, 82.94  , 84.2542, 79.9576, 76.34  , 71.9126,
       87.6942, 63.1546, 88.38  , 88.38  , 82.94  , 77.1134, 79.64  ,
       85.3088, 46.4416, 76.7296, 61.44  , 94.18  , 51.948 , 73.2008,
       88.56  , 88.2   , 39.7652, 70.3804, 72.3316, 82.94  , 73.04  ,
       82.76  , 52.05  , 55.9324, 50.4064, 81.3022, 82.7672, 82.5818,
       75.5998, 84.9142, 88.2314, 84.3254, 79.6944, 82.589 , 69.6356,
       72.2282, 76.3382, 72.06  , 82.904 , 83.6892, 64.234 , 77.7042,
       57.8894, 82.04  , 60.507 , 77.1896, 77.8802, 60.3722, 66.44  ,
       79.4546, 78.9104, 88.9522, 72.3236, 73.037 , 73.6468, 44.1516,
       67.2082, 94.18  , 82.94  , 89.2912, 62.1204, 88.2   , 81.7718,
       88.6708, 61.1534, 70.993 , 72.86  , 88.2   , 76.8736, 82.76  ,
       82.94  , 79.64  , 71.8094, 82.1752, 70.6592, 62.8564, 84.3474,
       66.2842, 82.04  , 72.0582, 76.4704, 82.3664, 88.5492, 76.3382,
       77.8896, 88.5006, 50.0118, 49.4036, 84.8784, 61.3446, 72.06  ,
       84.0292, 69.6906, 60.316 , 50.0658, 79.6944, 94.18  , 82.76  ,
       64.0072, 71.88  , 82.7654, 82.94  , 39.7126, 66.0854, 88.0236,
       89.991 , 33.1704, 77.3398, 88.38  , 93.2726, 79.64  , 67.4514,
       70.4576, 68.9696, 66.44  , 78.8456, 44.6632, 33.4714, 76.1798,
       85.1912, 83.7   , 83.7   , 70.943 , 78.4386, 82.4178, 72.3496,
       88.8744, 66.6438, 45.2442, 66.233 , 73.04  , 93.3268, 55.754 ,
       82.76  , 59.8414, 89.4518, 67.0328, 77.6662, 59.3912, 79.64  ,
```

```
       65.6396, 61.8126, 88.5688, 76.34  , 83.3832, 83.4588, 74.4064,
       76.7254, 87.6636, 82.98  , 72.2982, 78.7688, 82.76  , 30.9866])
```

In [37]:

```
from sklearn import metrics
```

In [38]:

```
print('MAE: ',metrics.mean_absolute_error(y_test, y_pred))
print('MSE: ',metrics.mean_squared_error(y_test, y_pred))
print('RMSE: ',np.sqrt(metrics.mean_absolute_error(y_test, y_pred)))
```

```
MAE:  0.8845729323308585
MSE:  5.087236515388475
RMSE:  0.9405173748160416
```

In [39]:

```
metrics.r2_score(y_test, y_pred)
```

Out[39]:

```
0.9722694819035159
```

Saving the Model:

In [41]:

```
import pickle
pickle.dump(rfr,open('wqa_app.pkl', 'wb'))
```