1)Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
```

2) importing Dataset

In [2]:
```
dataset = pd.read_csv('datasets_compresive_strength_concrete.csv')
```

In [3]:
```
dataset
```

Out[3]:

	Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Concrete compressive strength
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30
...	...	...	...	...	...	...	...	...	...
1025	276.4	116.0	90.3	179.6	8.9	870.1	768.3	28	44.28
1026	322.2	0.0	115.6	196.0	10.4	817.9	813.4	28	31.18
1027	148.5	139.4	108.6	192.7	6.1	892.4	780.0	28	23.70
1028	159.1	186.7	0.0	175.6	11.3	989.6	788.9	28	32.77
1029	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28	32.40

1030 rows × 9 columns

3)Taking care of null values

In [4]:
```
dataset.isnull().any()
```

Out[4]:
```
Cement                          False
Blast Furnace Slag              False
Fly Ash                         False
Water                           False
Superplasticizer                False
Coarse Aggregate                False
Fine Aggregate                  False
Age                             False
Concrete compressive strength   False
dtype: bool
```

No null values

4)Splitting in X and Y

In [5]:
```
x = dataset.iloc[:,0:8].values
y = dataset.iloc[:,8:9].values
```

In [6]:
```
x.shape
```

Out[6]:
```
(1030, 8)
```

In [7]:
```
y.shape
```

Out[7]:
```
(1030, 1)
```

5) Splitting into Train and Test

In [8]:
```
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,ran
dom_state =0)
```

In [9]:
```
x_train.shape
```

Out[9]: (824, 8)

In [10]:
```
y_train.shape
```

Out[10]: (824, 1)

In [11]:
```
x_test.shape
```

Out[11]: (206, 8)

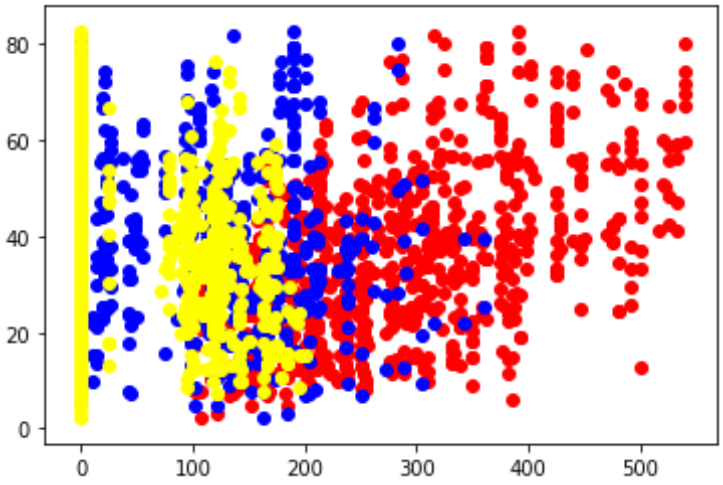
In [12]:
```
y_test.shape
```

Out[12]: (206, 1)

6)Data Visualization

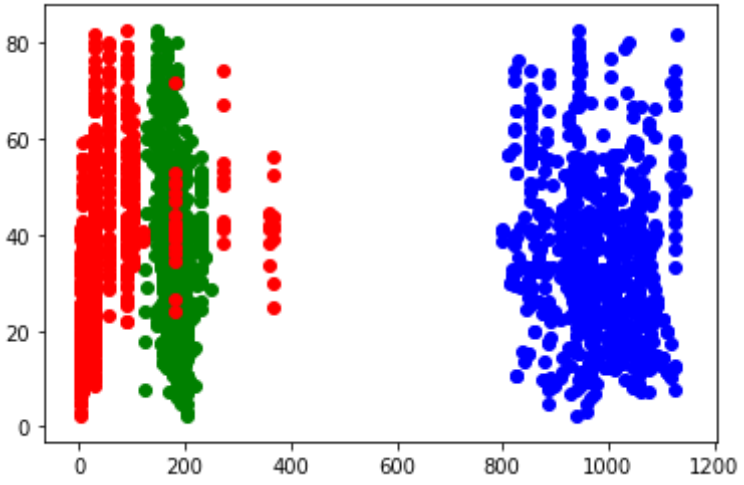
In [13]:
```
import matplotlib.pyplot as plt
plt.scatter(x_train[:,0],y_train[:,0],color = "red")
plt.scatter(x_train[:,1],y_train[:,0],color = "blue")
plt.scatter(x_train[:,2],y_train[:,0],color = "yellow")
```

Out[13]: <matplotlib.collections.PathCollection at 0x1aa76414ac8>



In [14]:
```
plt.scatter(x_train[:,3],y_train[:,0],color = "green")
plt.scatter(x_train[:,5],y_train[:,0],color = "blue")
plt.scatter(x_train[:,7],y_train[:,0],color = "red")
```

Out[14]: <matplotlib.collections.PathCollection at 0x1aa77494b48>



7)Applying Regression algorithm(Random

Forest Regressor)

In [15]:
```
from sklearn.ensemble import RandomForestRegressor
rfg = RandomForestRegressor(n_estimators= 20, random_state = 0)
rfg.fit(x_train,y_train)
```

```
C:\Users\vijay\anaconda3\lib\site-packages\ipykernel_launcher.py:3: Dat
aConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples,), for example using
ravel().
  This is separate from the ipykernel package so we can avoid doing imp
orts until
```

Out[15]:
```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nod
es=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.
0,
                      n_estimators=20, n_jobs=None, oob_score=False,
                      random_state=0, verbose=0, warm_start=False)
```

8)Saving the Model

In [16]:
```
import pickle
pickle.dump(rfg,open('concrete.pkl','wb'))
model=pickle.load(open('concrete.pkl','rb'))
```

In [17]:
```
x_train
```

Out[17]:
```
array([[ 480. ,    0. ,    0. , ...,  936. ,  721. ,   28. ],
       [ 375. ,    0. ,    0. , ..., 1038. ,  758. ,   28. ],
       [ 303.6,  139.9,    0. , ...,  895.5,  722.5,   28. ],
       ...,
       [ 144. ,    0. ,  175. , ...,  943. ,  844. ,   28. ],
```

```
                [ 239.6,  359.4,    0. , ...,  941.6,  664.3,   28. ],
                [ 192. ,  288. ,    0. , ...,  929.8,  716.1,   90. ]])
```

In [18]:
```
yrfg = rfg.predict(x_test)
```

In [19]:
```
yrfg
```

Out[19]:
```
array([19.349     ,  7.586     , 79.3       , 59.2605    , 11.497     ,
       48.19691667, 60.2525    , 21.827     , 73.752     , 51.381     ,
       18.428     , 42.6305    , 34.897     , 13.447     , 56.706     ,
       55.661     , 36.9055    , 34.775     , 55.828     , 39.014     ,
       57.36325   , 28.359     , 26.072     , 40.79333333, 16.986     ,
       25.38      , 62.937     , 25.27      , 55.661     , 59.6455    ,
       18.421     , 45.905     , 28.441     , 40.5615    , 21.2495    ,
        7.4805    , 34.328     , 26.1225    , 34.9235    , 33.004     ,
       37.185     , 33.9415    , 31.3805    , 39.792     , 57.44      ,
       33.545     , 27.837875  , 34.0995    , 33.2275    , 49.8465    ,
       44.351     , 25.5595    , 21.906     , 37.041     , 52.857     ,
       45.4285    , 44.1915    , 53.9175    , 64.1445    , 38.048     ,
       36.1305    , 23.406     , 62.57858333, 46.201     , 11.871     ,
       56.624     , 38.1235    , 25.895     , 13.892     , 15.238     ,
       15.8355    , 29.207     , 49.08      , 25.6545    , 38.984     ,
       30.9138    , 36.914     , 13.252     , 33.363     , 39.619     ,
       18.8235    , 50.279     , 18.1985    , 32.3965    , 47.5905    ,
       36.935     , 39.04893333, 53.498     , 41.735     , 38.4495    ,
       57.892     , 15.161     , 13.4435    , 40.1735    , 16.627     ,
       40.3365    , 15.879     , 40.718     , 49.3895    , 32.4075    ,
       16.859     , 19.0725    , 30.0855    , 27.9055    , 42.232     ,
       42.6035    , 52.013     , 15.6745    , 34.1105    , 46.378     ,
        8.6145    , 47.4225    , 14.7975    , 33.1365    , 42.795     ,
       59.481     , 37.1765    , 36.0945    , 34.48      , 17.5275    ,
       46.55      , 26.8365    , 65.27      , 28.525     , 12.7125    ,
       10.9325    , 11.5335    ,  8.216     , 53.054     , 37.5195    ,
       33.8175    , 53.1825    , 35.6045    , 29.783     , 13.5155    ,
       54.341     , 36.5335    , 44.094     ,  5.1265    , 12.7335    ,
       35.053     , 51.831     , 23.72      , 28.6795    , 18.1105    ,
       33.5055    , 39.48893333, 12.2795    , 41.339     , 18.064     ,
       14.7145    , 15.941     , 21.2015    , 38.278     , 39.0795    ,
       36.339     , 47.5065    , 27.945875  , 33.613     , 27.996     ,
```

```
41.333     , 40.124     , 14.6655    , 33.2305    , 30.9625    ,
 8.298     , 49.836     , 49.9035    , 52.0905    ,  9.8225    ,
36.538     , 37.4035    , 32.5205    , 34.8805    , 12.524     ,
13.8185    , 10.8635    , 40.4029    , 33.4995    , 53.6595    ,
15.157     , 28.3185    , 14.946     , 38.9875    , 16.765     ,
43.7215    , 36.256     , 44.428     , 15.1195    , 62.57858333,
31.895     , 68.759     , 60.2985    , 33.5195    ,  5.413     ,
46.4155    , 34.3585    , 27.4555    , 36.869     , 67.9205    ,
22.541     , 27.87      , 35.3385    , 27.0675    , 38.1235    ,
10.345     ])
```

In [20]:
```
y_test
```

Out[20]:
```
array([[26.06],
       [10.35],
       [79.3 ],
       [74.99],
       [ 9.69],
       [47.1 ],
       [59.  ],
       [22.72],
       [61.89],
       [52.12],
       [17.54],
       [48.15],
       [38.33],
       [17.2 ],
       [56.83],
       [55.25],
       [33.36],
       [34.68],
       [52.61],
       [39.94],
       [61.46],
       [27.63],
       [32.9 ],
       [41.64],
       [17.54],
       [26.85],
```

```
[66.9 ],
[21.06],
[61.07],
[66.95],
[20.87],
[48.79],
[24.05],
[47.81],
[21.16],
[ 6.94],
[28.6 ],
[26.31],
[33.95],
[25.72],
[37.8 ],
[35.17],
[32.24],
[37.81],
[57.23],
[33.4 ],
[30.14],
[33.8 ],
[35.08],
[52.91],
[40.93],
[21.75],
[16.5 ],
[42.35],
[50.94],
[64.02],
[52.2 ],
[53.39],
[64.3 ],
[44.52],
[37.42],
[22.14],
[62.94],
[45.7 ],
[14.64],
```

```
[53.58],
[33.72],
[24.58],
[13.2 ],
[18.91],
[ 7.4 ],
[24.85],
[49.2 ],
[25.57],
[31.38],
[23.35],
[45.71],
[ 6.81],
[27.34],
[39.59],
[30.96],
[55.64],
[18.02],
[33.09],
[48.72],
[36.8 ],
[45.9 ],
[46.68],
[40.87],
[35.34],
[65.7 ],
[17.24],
[ 9.45],
[33.94],
[15.57],
[40.27],
[15.82],
[41.54],
[50.24],
[29.98],
[20.73],
[27.42],
[27.66],
[29.73],
```

```
[45.08],
[42.13],
[51.96],
[15.61],
[34.29],
[53.52],
[ 4.83],
[41.37],
[ 9.73],
[31.81],
[41.05],
[61.92],
[37.42],
[37.92],
[32.72],
[14.94],
[44.09],
[26.92],
[64.9 ],
[29.22],
[12.46],
[ 8.  ],
[13.71],
[13.66],
[63.14],
[37.91],
[37.4 ],
[55.06],
[35.23],
[31.42],
[10.73],
[47.97],
[38.89],
[38.8 ],
[ 4.57],
[10.39],
[29.59],
[54.38],
[22.5 ],
```

```
[28.63],
[17.34],
[32.33],
[41.67],
[15.87],
[44.86],
[17.54],
[11.98],
[15.42],
[30.39],
[45.85],
[41.68],
[42.64],
[34.9 ],
[25.45],
[39.3 ],
[27.92],
[52.5 ],
[37.81],
[15.53],
[30.57],
[37.36],
[14.59],
[53.69],
[57.03],
[55.26],
[ 6.47],
[33.42],
[41.16],
[31.45],
[36.15],
[14.4 ],
[13.36],
[ 8.54],
[40.29],
[32.04],
[51.43],
[21.48],
[29.72],
```

```
       [14.2 ],
       [37.68],
       [24.13],
       [23.85],
       [36.45],
       [44.28],
       [19.52],
       [59.49],
       [25.18],
       [69.3 ],
       [61.23],
       [32.92],
       [ 6.28],
       [39.7 ],
       [33.4 ],
       [23.84],
       [39.27],
       [72.3 ],
       [18.13],
       [27.53],
       [33.76],
       [33.01],
       [33.72],
       [13.82]])
```

In [21]:
```
from sklearn.metrics import r2_score
accurfg =r2_score(y_test,yrfg)
```

In [22]:
```
accurfg
```

Out[22]:
```
0.9199600438446415
```

In [23]:
```
dataset.head(5)
```

Out[23]:

Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Concrete compressive strength

	Cement	Blast Furnace Slag	Fly Ash	Water	Superplasticizer	Coarse Aggregate	Fine Aggregate	Age	Concrete compressive strength
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30

In [24]:
```
p = rfg.predict([[234,234,78,457,6.8,1234,567,89]])
```

In [25]:
```
p[0]
```

Out[25]:
```
42.145
```

Done!

In [ ]:
```

```