In [72]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot
from sklearn.model_selection import KFold
```

In [73]:

```
df1 = pd.read_csv("Randomedi201.csv")
pd.set_option("display.max_columns", None)

df1
```

Out[73]:

	AnnuaL Mean Temperature	Precipitation Seasonality	Drainage Area (km^2)	Drainage Texture	Max Temperature of Warmest Month	Mean Annual Monsoon Flow (m3/s)
0	22.307884	93.547569	192.502640	0.000087	29.887011	11.945100
1	17.263815	67.755791	213.932037	0.000065	25.001167	1.656000
2	25.653601	112.179329	18450.912110	0.000634	33.946518	677.006000
3	23.128416	158.140106	420.320160	0.000274	34.606106	544.160318
4	24.650948	116.591377	5066.837402	0.000580	33.433296	206.518865
5	26.049732	141.446732	350.433624	0.000232	31.109547	138.724445
6	24.212206	131.164093	2306.040039	0.000326	38.008038	303.776635
7	23.686493	138.930984	3779.519775	0.000440	38.434139	215.481796
8	25.385981	118.453384	35702.257810	0.000263	34.655426	1231.582982
9	24.259298	115.949387	14968.109380	0.000361	38.112904	1254.719695
10	24.121849	131.046005	1942.562744	0.000220	35.905399	348.717909
11	25.424381	156.476242	791.283936	0.000142	36.807240	99.317412
12	25.237150	148.410629	541.287476	0.000218	35.212067	144.545187
13	24.768211	88.936455	69253.421880	0.000243	34.591045	346.094686
14	24.924210	98.838333	7870.645508	0.000265	36.391060	389.925116
15	25.415009	146.423004	300.548340	0.000035	33.123108	115.472911
16	25.485306	113.687416	15810.197270	0.000340	35.523151	666.938000
17	25.589821	134.401764	35479.253910	0.000129	34.859615	1165.876018
18	24.263506	112.860062	12061.231450	0.000540	41.233913	852.967704
19	26.207943	79.652954	8584.285156	0.000360	35.699432	13.793879
20	25.652811	100.314293	4394.541504	0.000590	42.526558	239.058909
21	24.182590	154.482971	993.997376	0.000620	41.296951	369.611892
22	27.012070	126.736610	45774.484380	0.000294	37.217041	1107.928515
23	24.751366	86.637672	2481.376465	0.000414	42.006042	19.657105
24	26.607559	124.961136	55306.183590	0.000246	34.648865	877.346462
25	24.400270	116.571381	2209.014648	0.000074	33.581619	82.910593
26	23.448601	136.135284	2821.115967	0.000119	34.769363	499.020634
27	24.699221	120.672516	6936.809570	0.000156	34.170368	259.925500
28	24.798716	96.240807	9891.224609	0.000116	33.098171	49.467920
29	25.451561	97.130249	9015.836914	0.000040	33.078529	166.651071
30	24.077030	91.857086	52122.167970	0.000156	32.179371	388.315564
31	25.490677	97.686272	66454.703130	0.000189	35.103558	383.739468
32	26.961117	124.130837	15843.124020	0.000176	32.562328	196.535595
33	25.921089	97.874069	6941.307617	0.000247	35.507923	154.396667
34	21.879065	141.480133	574.843384	0.000129	30.718559	92.548251

	Annual Mean Temperature	Precipitation Seasonality	Drainage Area (km^2)	Drainage Texture	Max Temperature of Warmest Month	Mean Annual Monsoon Flow (m3/s)
35	24.137991	107.703888	14475.829130	0.000380	34.072067	614.985385
36	24.777304	99.339409	6891.486816	0.000174	32.926483	174.966434
37	25.242662	137.147522	3266.582031	0.000762	35.860931	982.445285
38	25.401436	97.490074	64073.714840	0.000126	37.038273	432.047007
39	22.698811	80.948708	5579.921387	0.000252	35.163727	21.277907
40	24.740263	109.545830	8494.139648	0.000749	35.971535	391.173908

In [74]:

```
#Preparing data for training
X = df1.iloc[:, 0:5].values
y = df1.iloc[:, 5].values
```

In [75]:

```
y
```

Out[75]:

```
array([  11.9451    ,    1.656     ,  677.006     ,  544.1603185 ,
        206.5188645 ,  138.7244453 ,  303.7766345 ,  215.4817955 ,
       1231.582982  , 1254.719695  ,  348.7179089 ,   99.31741223,
        144.5451866 ,  346.0946862 ,  389.9251163 ,  115.472911  ,
        666.938     , 1165.876018  ,  852.9677039 ,   13.79387901,
        239.0589089 ,  369.6118921 , 1107.928515  ,   19.65710532,
        877.3464618 ,   82.91059349,  499.0206337 ,  259.9255    ,
         49.46792048,  166.6510707 ,  388.3155636 ,  383.7394676 ,
        196.5355946 ,  154.3966668 ,   92.54825115,  614.9853847 ,
        174.9664336 ,  982.4452852 ,  432.0470074 ,   21.27790745,
        391.1739084 ])
```

In [76]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test= train_test_split(X,y, test_size = 0.20, random_state = 0)
```

In [77]:

```
kf = KFold(n_splits=2, random_state = 0)
kf.get_n_splits(X)
```

Out[77]:

```
2
```

In [78]:

```
for train_index, test_index in kf.split(X):
    print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
```

```
TRAIN: [21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40] TEST: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 1
9 20]
TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20] TEST: [21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 3
9 40]
```

In [79]:

```
#Feature scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

In [80]:

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators =12, random_state = 0)
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

In [81]:

```
y_pred
```

Out[81]:

```
array([ 339.68144187, 1157.45992975,  216.46450227, 1106.55006242,
        173.07160912,  212.85398298,  254.53780443,  156.9970598 ,
        127.33703567,  615.75704448,  524.53581843,  358.70837872,
        225.82162573,  253.58050782,  290.28000167,  168.69291525,
        221.77878346,  600.25456916,  191.91561303,  249.70319971])
```

In [50]:

```
df5 = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
df5.to_csv("MonsoonFlow02.csv")
df5
```

Out[50]:

	Real Values	Predicted Values
0	369.611892	339.681442
1	1107.928515	1157.459930
2	19.657105	216.464502
3	877.346462	1106.550062
4	82.910593	173.071609
5	499.020634	212.853983
6	259.925500	254.537804
7	49.467920	156.997060
8	166.651071	127.337036
9	388.315564	615.757044
10	383.739468	524.535818
11	196.535595	358.708379
12	154.396667	225.821626
13	92.548251	253.580508
14	614.985385	290.280002
15	174.966434	168.692915
16	982.445285	221.778783
17	432.047007	600.254569
18	21.277907	191.915613
19	391.173908	249.703200

In [82]:

```
regressor.score(X_test,y_test)
```

Out[82]:

```
0.436813894944035
```

In [83]:

```
a=metrics.mean_absolute_error(y_test,y_pred)
b=metrics.mean_squared_error(y_test,y_pred)
c=np.sqrt(metrics.mean_absolute_error(y_test,y_pred))
```

```
c=np.sqrt(metrics.mean_absolute_error(y_test,y_pred))
a,b,c
```

Out[83]:

(168.4430304694583, 54408.58687709363, 12.978560415911247)

In [84]:

```
y_pred1 = regressor.predict(X_train)
```

In [85]:

```
y_pred1
```

Out[85]:

```
array([  69.79298543,   64.95656527,  689.15247533,  365.50442156,
        206.10588982,  115.34150606,  304.0412528 ,  244.29880673,
       1124.90486283, 1254.719695  ,  276.8913536 ,  203.24711278,
        145.40725727,  533.86821603,  262.32944548,  106.09063815,
        705.52434042, 1090.12332142,  846.68200029,  187.00320129,
        239.71577573])
```

In [55]:

```
df6 = pd.DataFrame({'Real Values':y_train, 'Predicted Values':y_pred1})
df6.to_csv("monsoonflow01.csv")
df6
```

Out[55]:

	Real Values	Predicted Values
0	11.945100	69.792985
1	1.656000	64.956565
2	677.006000	689.152475
3	544.160318	365.504422
4	206.518865	206.105890
5	138.724445	115.341506
6	303.776635	304.041253
7	215.481796	244.298807
8	1231.582982	1124.904863
9	1254.719695	1254.719695
10	348.717909	276.891354
11	99.317412	203.247113
12	144.545187	145.407257
13	346.094686	533.868216
14	389.925116	262.329445
15	115.472911	106.090638
16	666.938000	705.524340
17	1165.876018	1090.123321
18	852.967704	846.682000
19	13.793879	187.003201
20	239.058909	239.715776

In [86]:

```
from sklearn import metrics
a1=metrics.mean_absolute_error(y_train,y_pred1)
b1=metrics.mean_squared_error(y_train,y_pred1)
c1=np.sqrt(metrics.mean_absolute_error(y_train,y_pred1))
```

```
c1=np.sqrt(metrics.mean_absolute_error(y_train,y_pred1))
a1,b1,c1
```

Out[86]:

(60.350819777658714, 7477.8094121881195, 7.7685790063343445)

In [87]:

```
regressor.score(X_train,y_train)
```

Out[87]:

0.9515745894287226

In [88]:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectFromModel
```

In [89]:

```
importance=regressor.feature_importances_
```

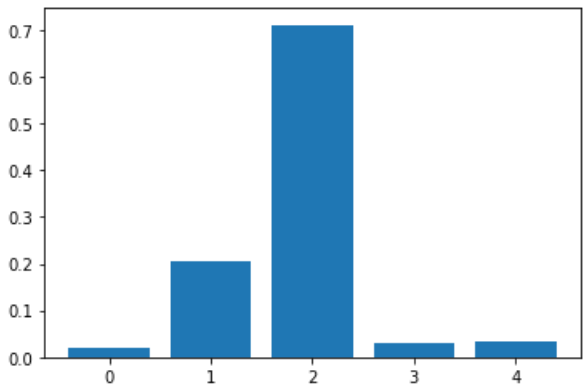
In [90]:

```
from sklearn.datasets import make_regression
from sklearn.ensemble import RandomForestRegressor
```

In [91]:

```
# summarize feature importance
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
pyplot.show()
```

```
Feature: 0, Score: 0.02031
Feature: 1, Score: 0.20408
Feature: 2, Score: 0.71054
Feature: 3, Score: 0.03014
Feature: 4, Score: 0.03494
```



In [92]:

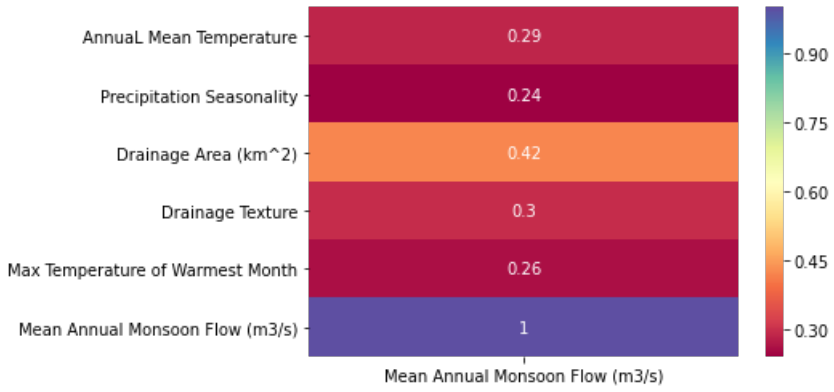
```
#Correlation Matrix

corr = df1.corr()
corr.style.background_gradient(cmap='RdYlGn')
```

Out[92]:

	AnnuaL Mean Temperature	Precipitation Seasonality	Drainage Area (km^2)	Drainage Texture	Max Temperature of Warmest Month	Mean Annual Menseen Flow (m3/s)
AnnuaL Mean Temperature	1	0.216688	0.325187	0.178263	0.430429	0.287834
Precipitation Seasonality	0.216688	1	-0.262082	0.0888714	0.175863	0.241565
Drainage Area (km^2)	0.325187	-0.262082	1	-0.138631	0.0174227	0.422622
Drainage Texture	0.178263	0.0888714	-0.138631	1	0.549694	0.296492
Max Temperature of Warmest Month	0.430429	0.175863	0.0174227	0.549694	1	0.261624
Mean Annual Monsoon Flow (m3/s)	0.287834	0.241565	0.422622	0.296492	0.261624	1

In [96]:

```
import seaborn as sns
x = corr[['Mean Annual Monsoon Flow (m3/s)']]
sns.heatmap(x,annot=True,cmap="Spectral")
pyplot.savefig("aa4.png", dpi=400, bbox_inches='tight')
```



In [ ]:

```
```

In [ ]:

```
```