In [1]:
```
import pandas as pd
df = pd.read_csv("HDI.csv")
```

In [2]:
```
df.head()
```

Out[2]:

	iso3	country	hdicode	region	hdi_rank_2021	hdi_1990	hdi_1991	hdi_1992	hdi_1
0	AFG	Afghanistan	Low	SA	180.0	0.273	0.279	0.287	0
1	AGO	Angola	Medium	SSA	148.0	NaN	NaN	NaN	
2	ALB	Albania	High	ECA	67.0	0.647	0.629	0.614	0
3	AND	Andorra	Very High	NaN	40.0	NaN	NaN	NaN	
4	ARE	United Arab Emirates	Very High	AS	26.0	0.728	0.739	0.742	0

5 rows × 1008 columns

In [3]:
```
df.describe()
```

Out[3]:

	hdi_rank_2021	hdi_1990	hdi_1991	hdi_1992	hdi_1993	hdi_1994	hdi_19
count	191.000000	152.000000	152.000000	152.000000	152.000000	152.000000	163.000
mean	95.811518	0.595112	0.597862	0.600493	0.604474	0.609329	0.613
std	55.307333	0.161918	0.161921	0.162193	0.163122	0.163818	0.162
min	1.000000	0.216000	0.218000	0.222000	0.227000	0.232000	0.238
25%	48.500000	0.477750	0.477000	0.475250	0.474250	0.476500	0.480
50%	96.000000	0.621500	0.623500	0.622000	0.624000	0.623500	0.642
75%	143.500000	0.725500	0.727000	0.723750	0.724250	0.733750	0.737
max	191.000000	0.872000	0.873000	0.878000	0.880000	0.884000	0.885

8 rows × 1004 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206 entries, 0 to 205
Columns: 1008 entries, iso3 to mf_2021
dtypes: float64(1004), object(4)
memory usage: 1.6+ MB
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
iso3              0
country           0
hdicode          15
region           55
hdi_rank_2021    15
                 ..
mf_2017          38
mf_2018          38
mf_2019          38
mf_2020          38
mf_2021          38
Length: 1008, dtype: int64
```

In [12]:
```
nums_cols = df.select_dtypes(include=["number"]).columns
obj_cols = df.select_dtypes(include=["object"]).columns
```

In [13]:
```
from sklearn.impute import SimpleImputer
```

In [16]:
```
nums_imputer = SimpleImputer(strategy="median")
df[nums_cols] = nums_imputer.fit_transform(df[nums_cols])
```

In [17]:
```
obj_imputer = SimpleImputer(strategy="most_frequent")
df[obj_cols] = obj_imputer.fit_transform(df[obj_cols])
```

In [19]:
```
df.head()
```

Out[19]:

	iso3	country	hdicode	region	hdi_rank_2021	hdi_1990	hdi_1991	hdi_1992	hdi_1
0	AFG	Afghanistan	Low	SA	180.0	0.2730	0.2790	0.287	0
1	AGO	Angola	Medium	SSA	148.0	0.6215	0.6235	0.622	0
2	ALB	Albania	High	ECA	67.0	0.6470	0.6290	0.614	0
3	AND	Andorra	Very High	SSA	40.0	0.6215	0.6235	0.622	0
4	ARE	United Arab Emirates	Very High	AS	26.0	0.7280	0.7390	0.742	0

5 rows × 1008 columns

In [23]:
```
X = df.drop(columns=["iso3","country","hdicode","region"]) # these values may confu
y = df["hdicode"]
```

In [24]:
```
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=42
)
```

In [26]:
```
#Standardization and normalization is not required in random forest so,we can't per
from sklearn.ensemble import RandomForestClassifier
```

In [28]:
```
rf = RandomForestClassifier(
    n_estimators = 500,
    random_state = 42
)
```

In [29]:
```
rf.fit(X_train,y_train)
```

Out[29]:
```
RandomForestClassifier(n_estimators=500, random_state=42)
```

In [32]:
```
y_pred = rf.predict(X_test)
y_pred
```

Out[32]:
```
array(['Medium', 'Very High', 'High', 'Very High', 'Very High', 'High',
       'Low', 'Very High', 'Very High', 'Very High', 'Very High',
       'Very High', 'Low', 'Very High', 'Medium', 'High', 'Very High',
       'Medium', 'High', 'Medium', 'High', 'Very High', 'High',
       'Very High', 'Very High', 'Very High', 'Medium', 'Very High',
       'High', 'Very High', 'Medium', 'Very High', 'Medium', 'Low',
       'Medium', 'High', 'Very High', 'Low', 'Low', 'Very High',
       'Very High', 'Low'], dtype=object)
```

In [35]:
```
y_test.head()
```

Out[35]:
```
15        Medium
9      Very High
201    Very High
82     Very High
68     Very High
Name: hdicode, dtype: object
```

In [40]:
```
from sklearn.metrics import accuracy_score,classification_report
accu = accuracy_score(y_pred,y_test)
accu
```

Out[40]:
```
0.9285714285714286
```

In [44]:
```
cr = classification_report(y_pred,y_test)
print("classification report\n",cr)
```

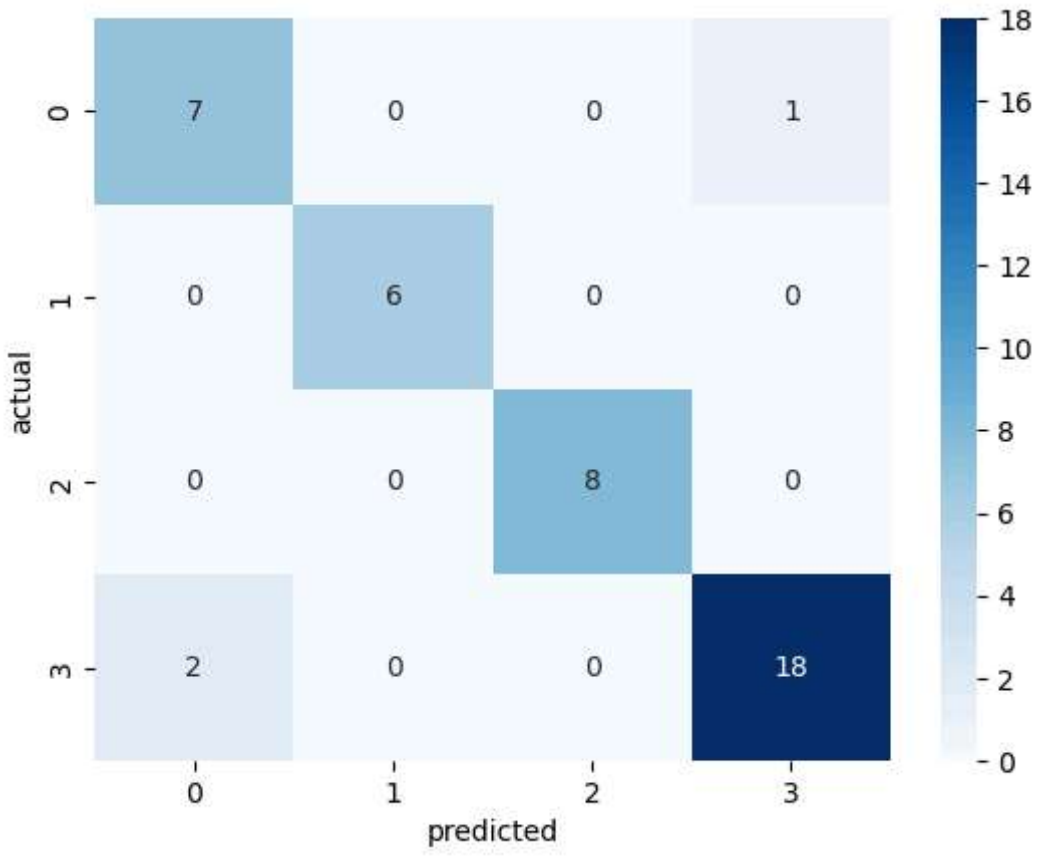
```
classification report
              precision    recall  f1-score   support

        High       0.78      0.88      0.82         8
         Low       1.00      1.00      1.00         6
      Medium       1.00      1.00      1.00         8
   Very High       0.95      0.90      0.92        20

    accuracy                           0.93        42
   macro avg       0.93      0.94      0.94        42
weighted avg       0.93      0.93      0.93        42
```

In [46]:
```
from sklearn.metrics import confusion_matrix # we make heatmap for confusion matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

In [47]:
```
cm = confusion_matrix(y_pred,y_test)
```

In [58]:
```
sns.heatmap(cm,annot=True,fmt="d",cmap="Blues")
plt.xlabel("predicted")
plt.ylabel("actual")
```

Out[58]: Text(50.722222222222214, 0.5, 'actual')



In [59]:
```
import pandas as pd

feature_importance = pd.Series(
    rf.feature_importances_,
```

```
    index=X_train.columns
).sort_values(ascending=False)

print(feature_importance.head(10))
```

```
hdi_rank_2021    0.034782
hdi_2021         0.033009
hdi_2018         0.024834
hdi_2020         0.022574
hdi_2019         0.017154
hdi_2014         0.016128
hdi_2016         0.015040
hdi_2013         0.014717
hdi_2017         0.014099
hdi_2015         0.013302
dtype: float64
```