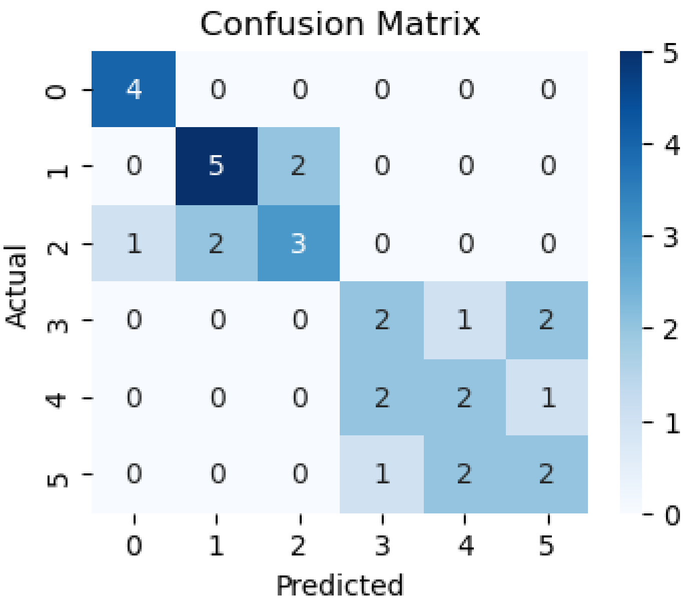
In [6]:

```python
# KNN

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd

from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_sco
from sklearn.model_selection import train_test_split
df=pd.read_csv("btissue.csv")
# print(df)
X=df.iloc[:, :-1].values
y=df.iloc[:, -1].values
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_stat
knn = KNeighborsClassifier(n_neighbors=3)
model=knn.fit(X_train,y_train)
y_pred=model.predict(X_test)

print("\n---KNN Results ---")
print("Accuracy:",accuracy_score(y_test,y_pred))
print("Precision:",precision_score(y_test,y_pred,average='macro',zero_divisior
print("Recall:",recall_score(y_test,y_pred,average='macro',zero_division=0))
print("F1-score:",f1_score(y_test,y_pred,average='macro',zero_division=0))

cm=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(4,3))
sns.heatmap(cm,annot=True,fmt='d',cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

```
---KNN Results ---
Accuracy: 0.5625
Precision: 0.5523809523809523
Recall: 0.569047619047619
F1-score: 0.5581048581048581
```

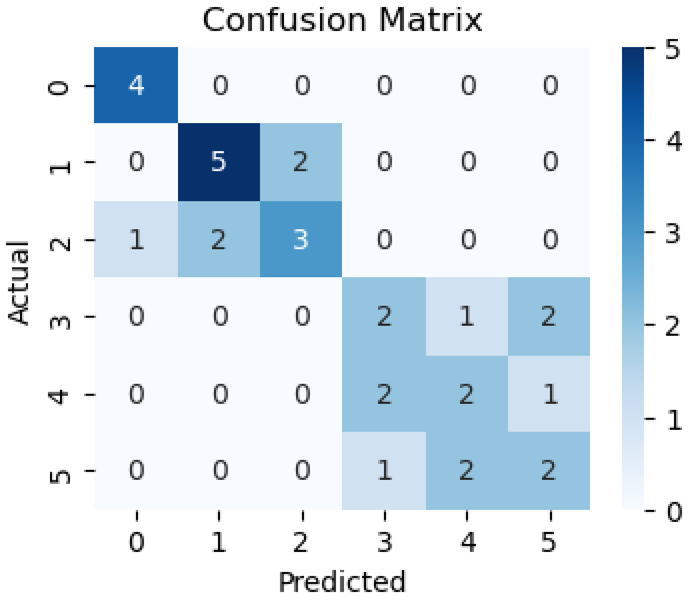


	0	1	2	3	4	5
0	4	0	0	0	0	0
1	0	5	2	0	0	0
2	1	2	3	0	0	0
3	0	0	0	2	1	2
4	0	0	0	2	2	1
5	0	0	0	1	2	2

In [5]:

```
# Decision Tree Model

import matplotlib.pyplot as plt
import seaborn as sns

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_scc
from sklearn.model_selection import train_test_split
df=pd.read_csv("btissue.csv")
dt=DecisionTreeClassifier(max_depth=5,random_state=42)
#print(df)
X=df.iloc[:, :-1].values
y=df.iloc[:, -1].values
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_stat
model=dt.fit(X_train,y_train)
y_pred_dt=model.predict(X_test)

print("\n---Decision Tree Results ---")
print("Accuracy:",accuracy_score(y_test,y_pred))
print("Precision:",precision_score(y_test,y_pred,average='macro',zero_divisior
print("Recall:",recall_score(y_test,y_pred,average='macro',zero_division=0))
print("F1-score:",f1_score(y_test,y_pred,average='macro',zero_division=0))

cm=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(4,3))
sns.heatmap(cm,annot=True,fmt='d',cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

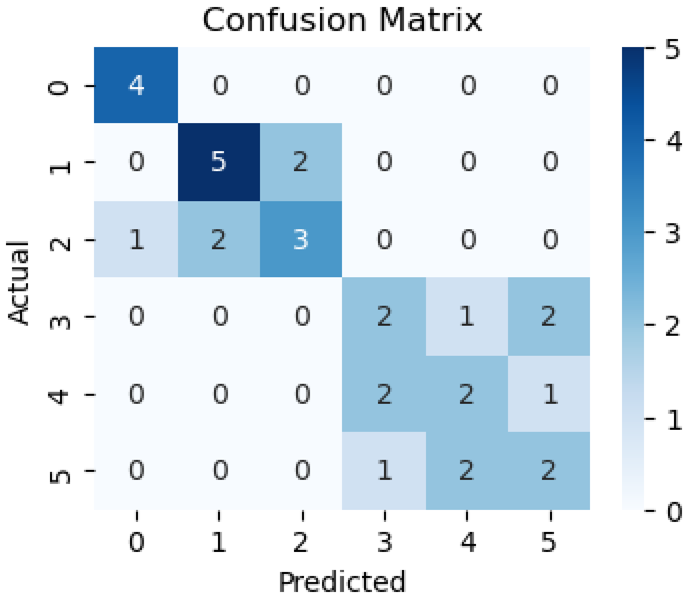
```
---Decision Tree Results ---
Accuracy: 0.5625
Precision: 0.5523809523809523
Recall: 0.569047619047619
F1-score: 0.5581048581048581
```



In [4]:

```
# Random Forest Model

import matplotlib.pyplot as plt
import seaborn as sns

import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_scc
from sklearn.model_selection import train_test_split
df=pd.read_csv("btissue.csv")
rf=RandomForestClassifier(max_depth=5,random_state=42)
# print(df)
X=df.iloc[:, :-1].values
y=df.iloc[:, -1].values
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_stat
model=rf.fit(X_train,y_train)
y_pred_dt=model.predict(X_test)

print("\n---Random Forest Results ---")
print("Accuracy:",accuracy_score(y_test,y_pred))
print("Precision:",precision_score(y_test,y_pred,average='macro',zero_divisior
print("Recall:",recall_score(y_test,y_pred,average='macro',zero_division=0))
print("F1-score:",f1_score(y_test,y_pred,average='macro',zero_division=0))

cm=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(4,3))
sns.heatmap(cm,annot=True,fmt='d',cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

```
---Random Forest Results ---
Accuracy: 0.5625
Precision: 0.5523809523809523
Recall: 0.569047619047619
F1-score: 0.5581048581048581
```

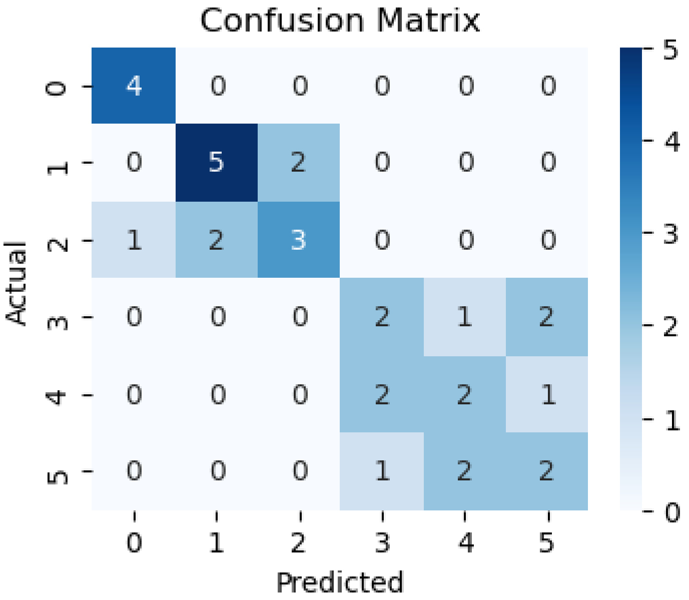


In [7]:

```
# SVM Model

import matplotlib.pyplot as plt
import seaborn as sns

import pandas as pd
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_sco
from sklearn.model_selection import train_test_split
df=pd.read_csv("btissue.csv")
svm =SVC(kernel='linear',random_state=12)
X=df.iloc[:, :-1].values
y=df.iloc[:, -1].values
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_stat
model=svm.fit(X_train,y_train)
y_pred_svm=model.predict(X_test)

print("\n---SVM Results ---")
print("Accuracy:",accuracy_score(y_test,y_pred))
print("Precision:",precision_score(y_test,y_pred,average='macro',zero_divisior
print("Recall:",recall_score(y_test,y_pred,average='macro',zero_division=0))
print("F1-score:",f1_score(y_test,y_pred,average='macro',zero_division=0))

cm=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(4,3))
sns.heatmap(cm,annot=True,fmt='d',cmap='Blues')
```

```
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

```
---SVM Results ---
Accuracy: 0.5625
Precision: 0.5523809523809523
Recall: 0.569047619047619
F1-score: 0.5581048581048581
```



Actual \ Predicted	0	1	2	3	4	5
0	4	0	0	0	0	0
1	0	5	2	0	0	0
2	1	2	3	0	0	0
3	0	0	0	2	1	2
4	0	0	0	2	2	1
5	0	0	0	1	2	2

```
In [ ]:
```