In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split, cross_val_predict, KFold
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, classification_report
```

In [3]:
```
iris = load_iris()
X = iris.data
y = iris.target
```

In [4]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

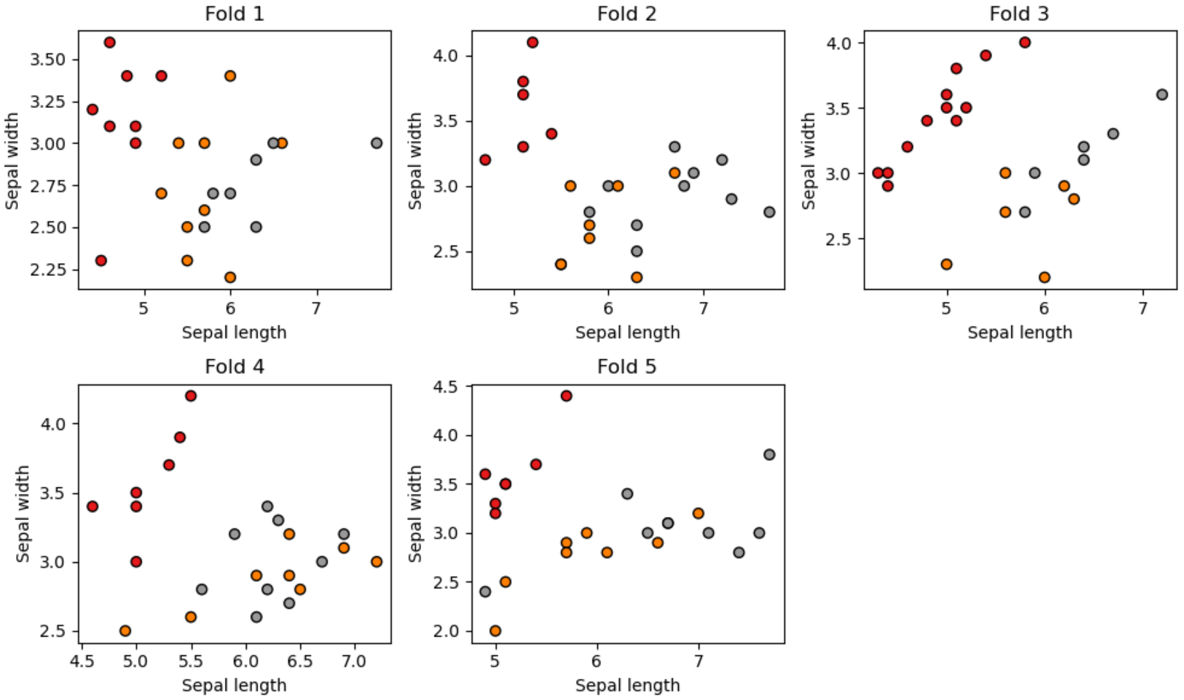
In [6]:
```
clf = DecisionTreeClassifier()
kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

In [7]:
```
plt.figure(figsize=(10, 6))
for i, (train_index, test_index) in enumerate(kf.split(X_train)):
    X_train_kf, X_val_kf = X_train[train_index], X_train[test_index]
    y_train_kf, y_val_kf = y_train[train_index], y_train[test_index]

    clf.fit(X_train_kf, y_train_kf)
    y_pred_kf = clf.predict(X_val_kf)

    plt.subplot(2, 3, i+1)
    plt.scatter(X_val_kf[:, 0], X_val_kf[:, 1], c=y_pred_kf, cmap=plt.cm.Set1, edge
    plt.xlabel('Sepal length')
    plt.ylabel('Sepal width')
    plt.title(f'Fold {i+1}')

plt.tight_layout()
plt.show()
```



In [8]:
```
y_pred_test = cross_val_predict(clf, X_test, y_test, cv=5)
conf_mat = confusion_matrix(y_test, y_pred_test)
class_report = classification_report(y_test, y_pred_test)

print("Confusion Matrix:")
print(conf_mat)
print("\nClassification Report:")
print(class_report)
```

```
Confusion Matrix:
[[10  0  0]
 [ 1  8  0]
 [ 0  1 10]]

Classification Report:
              precision    recall  f1-score   support

           0       0.91      1.00      0.95        10
           1       0.89      0.89      0.89         9
           2       1.00      0.91      0.95        11

    accuracy                           0.93        30
   macro avg       0.93      0.93      0.93        30
weighted avg       0.94      0.93      0.93        30
```

In [9]:

```
import warnings
warnings.filterwarnings('ignore')
from sklearn.datasets import load_boston
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
```

In [10]:

```
boston = load_boston()
X = boston.data
y = boston.target
```

In [11]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [12]:

```
regressor = DecisionTreeRegressor()
kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

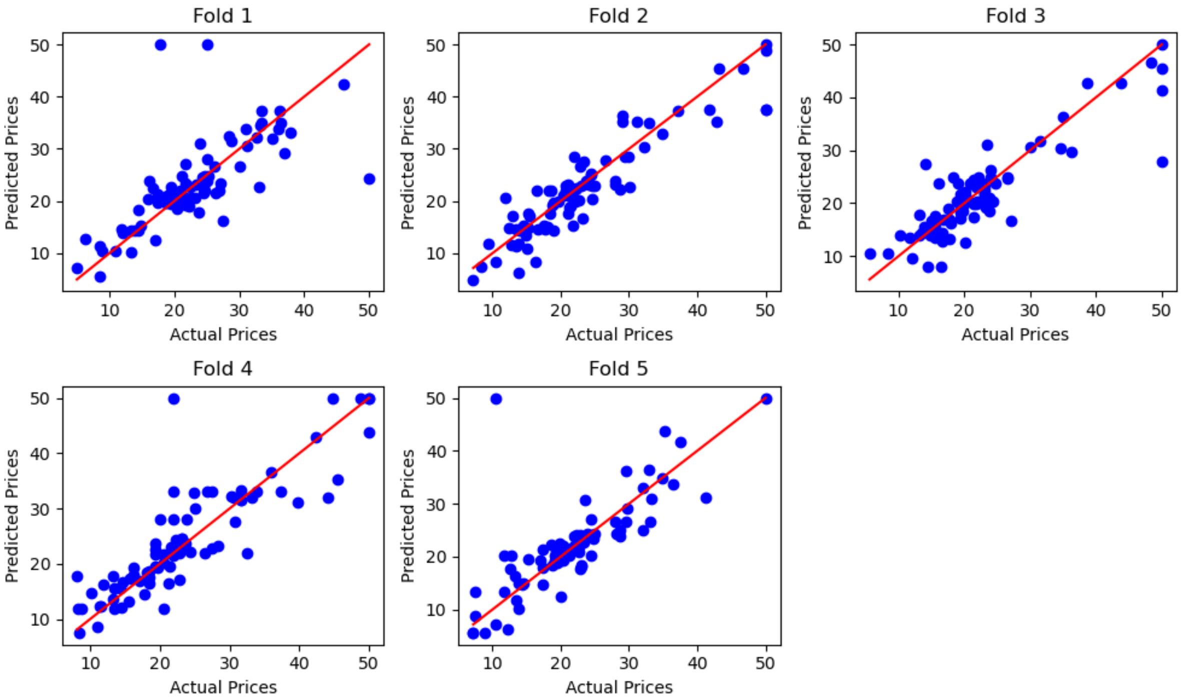
In [13]:

```
plt.figure(figsize=(10, 6))
for i, (train_index, test_index) in enumerate(kf.split(X_train)):
    X_train_kf, X_val_kf = X_train[train_index], X_train[test_index]
    y_train_kf, y_val_kf = y_train[train_index], y_train[test_index]

    regressor.fit(X_train_kf, y_train_kf)
    y_pred_kf = regressor.predict(X_val_kf)

    plt.subplot(2, 3, i+1)
    plt.scatter(y_val_kf, y_pred_kf, color='blue')
    plt.plot([min(y_val_kf), max(y_val_kf)], [min(y_val_kf), max(y_val_kf)], color=
    plt.xlabel('Actual Prices')
    plt.ylabel('Predicted Prices')
    plt.title(f'Fold {i+1}')

plt.tight_layout()
plt.show()
```



In [14]:

```
y_pred_test = cross_val_predict(regressor, X_test, y_test, cv=5)
mse = mean_squared_error(y_test, y_pred_test)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 27.744411764705884
```

In [ ]: