```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Load dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split into train and test sets
X_train,   X_test,   y_train,   y_test   =   train_test_split(X,   y,   test_size=0.3,
random_state=42)

# Normalize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# KNN classification for different K values
k_values = [1, 3, 5, 7]
results = []

for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    results.append((k, acc, cm))

# Confusion matrix for best K
best_k, best_acc, best_cm = max(results, key=lambda x: x[1])
plt.figure(figsize=(6, 4))
sns.heatmap(best_cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.title(f'Confusion Matrix for K={best_k}')
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

```
plt.tight_layout()
plt.savefig("/mnt/data/knn_confusion_matrix.png")
plt.close()
```