```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
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,
classification_report

def load_data():
    iris = load_iris()
    X = iris.data
    y = iris.target
    labels = iris.target_names
    return X, y, labels

def preprocess(X, y):
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
    return train_test_split(X_scaled, y, test_size=0.3, random_state=42)

def train_knn(X_train, y_train, k=3):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    return knn

def evaluate_model(model, X_test, y_test, labels):
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f"\nAccuracy: {acc:.2f}")
    print("\nClassification Report:\n", classification_report(y_test, y_pred,
target_names=labels))
    return y_test, y_pred, confusion_matrix(y_test, y_pred)

def visualize(cm, labels):
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=labels,
yticklabels=labels)
    plt.title("Confusion Matrix - KNN Iris Classifier")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

def main():
    print("=== Iris Flower Classification using KNN ===")
    X, y, labels = load_data()
    X_train, X_test, y_train, y_test = preprocess(X, y)
    model = train_knn(X_train, y_train, k=3)
    y_test_vals, y_pred_vals, cm = evaluate_model(model, X_test, y_test, labels)
    visualize(cm, labels)

if __name__ == "__main__":
    main()
```

5