```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression

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

def evaluate_model(name, model, X_train, X_test, y_train, y_test, target_names):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f"\n{name} Accuracy: {acc:.2f}")
    print(f"{name} Classification Report:\n", classification_report(y_test,
y_pred, target_names=target_names))
    return name, acc

def visualize_accuracies(results):
    models = [x[0] for x in results]
    scores = [x[1] for x in results]

    plt.figure()
    plt.bar(models, scores, color='skyblue')
    plt.ylim(0, 1)
    plt.title("Classifier Accuracy Comparison")
    plt.ylabel("Accuracy Score")
    plt.xlabel("Classifier")
    plt.grid(axis='y')
    plt.show()

def main():
    print("=== Classification Algorithms Comparison ===")
    X, y, target_names = load_data()
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

    classifiers = [
        ("K-Nearest Neighbors", KNeighborsClassifier(n_neighbors=3)),
        ("Naive Bayes", GaussianNB()),
        ("Decision Tree", DecisionTreeClassifier()),
        ("Logistic Regression", LogisticRegression(max_iter=200))
    ]

    results = []
    for name, model in classifiers:
        results.append(evaluate_model(name, model, X_train, X_test, y_train,
y_test, target_names))

    visualize_accuracies(results)

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