```python
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.naive_bayes import GaussianNB
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 split_data(X, y):
    return train_test_split(X, y, test_size=0.3, random_state=42)

def train_naive_bayes(X_train, y_train):
    model = GaussianNB()
    model.fit(X_train, y_train)
    return model

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

def visualize_confusion_matrix(cm, labels):
    sns.heatmap(cm, annot=True, fmt='d', cmap='Greens', xticklabels=labels,
yticklabels=labels)
    plt.title("Naive Bayes - Confusion Matrix")
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.show()

def main():
    print("=== Iris Flower Classification using Naive Bayes ===")
    X, y, labels = load_data()
    X_train, X_test, y_train, y_test = split_data(X, y)
    model = train_naive_bayes(X_train, y_train)
    cm = evaluate_model(model, X_test, y_test, labels)
    visualize_confusion_matrix(cm, labels)

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