```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from collections import Counter

iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

class KNN:
    def __init__(self, k=3):
        self.k = k

    def fit(self, X, y):
        self.X = X
        self.y = y

    def _predict(self, x):
        dists = np.array([np.linalg.norm(Xinst - x) for Xinst in self.X])
        k_indices = np.argsort(dists)[0:self.k]
        k_labels = [self.y[i] for i in k_indices]
        c = Counter(k_labels)
        return c.most_common(1)[0][0]

    def predict(self, X):
        return np.array([self._predict(x) for x in X])

knn = KNN(k=3)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
print(np.mean(y_pred == y_test))
```

```
1.0
```