

KNN

```
import numpy as np
from collections import Counter

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

    def fit(self, X_train, y_train):
        self.X_train = X_train
        self.y_train = y_train

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

    def _predict(self, x):
        # Compute distances to all training points
        distances = np.linalg.norm(self.X_train - x, axis=1)
        # Get indices of k nearest neighbors
        k_idx = np.argsort(distances)[:self.k]
        # Get the labels of the k nearest
        k_labels = self.y_train[k_idx]
        # Return the most common label
        return Counter(k_labels).most_common(1)[0][0]
```