

KNN

August 2, 2024

```
[3]: import numpy as np
from collections import Counter

def euclidean_distances(x1,x2):
    return np.sqrt(np.sum(x1-x2)**2)

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

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

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

    def _predict(self,x):
        # compute distances
        distances=[euclidean_distances(x,x_train) for x_train in self.X_train]
        # k nearest samples, labels
        k_indices=np.argsort(distances)[:self.k]
        k_nearest_labels=[self.y_train[i] for i in k_indices]
        # majority vote, most common class label
        most_common=Counter(k_nearest_labels).most_common(1)
        return most_common[0][0]
```

```
[4]: import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn import datasets
from sklearn.model_selection import train_test_split
import numpy as np

cmap = ListedColormap(["#FF0000", "#00FF00", "#0000FF"])
iris = datasets.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=1234)
```

```
from KNN import KNN  
clf=KNN(k=3)  
clf.fit(X_train,y_train)  
predictions=clf.predict(X_test)  
acc=np.sum(predictions==y_test)/len(y_test)  
print(acc)
```

0.9333333333333333