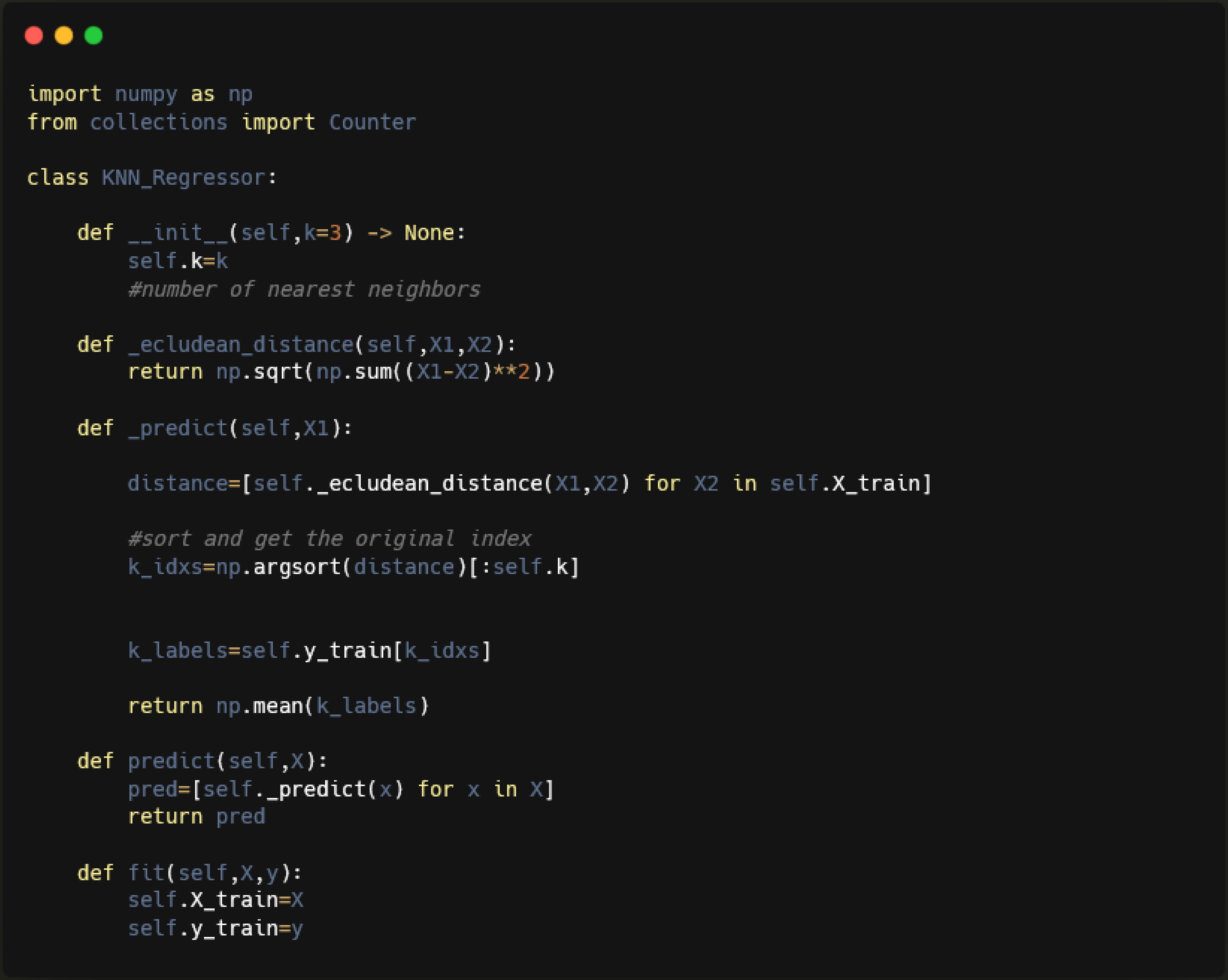
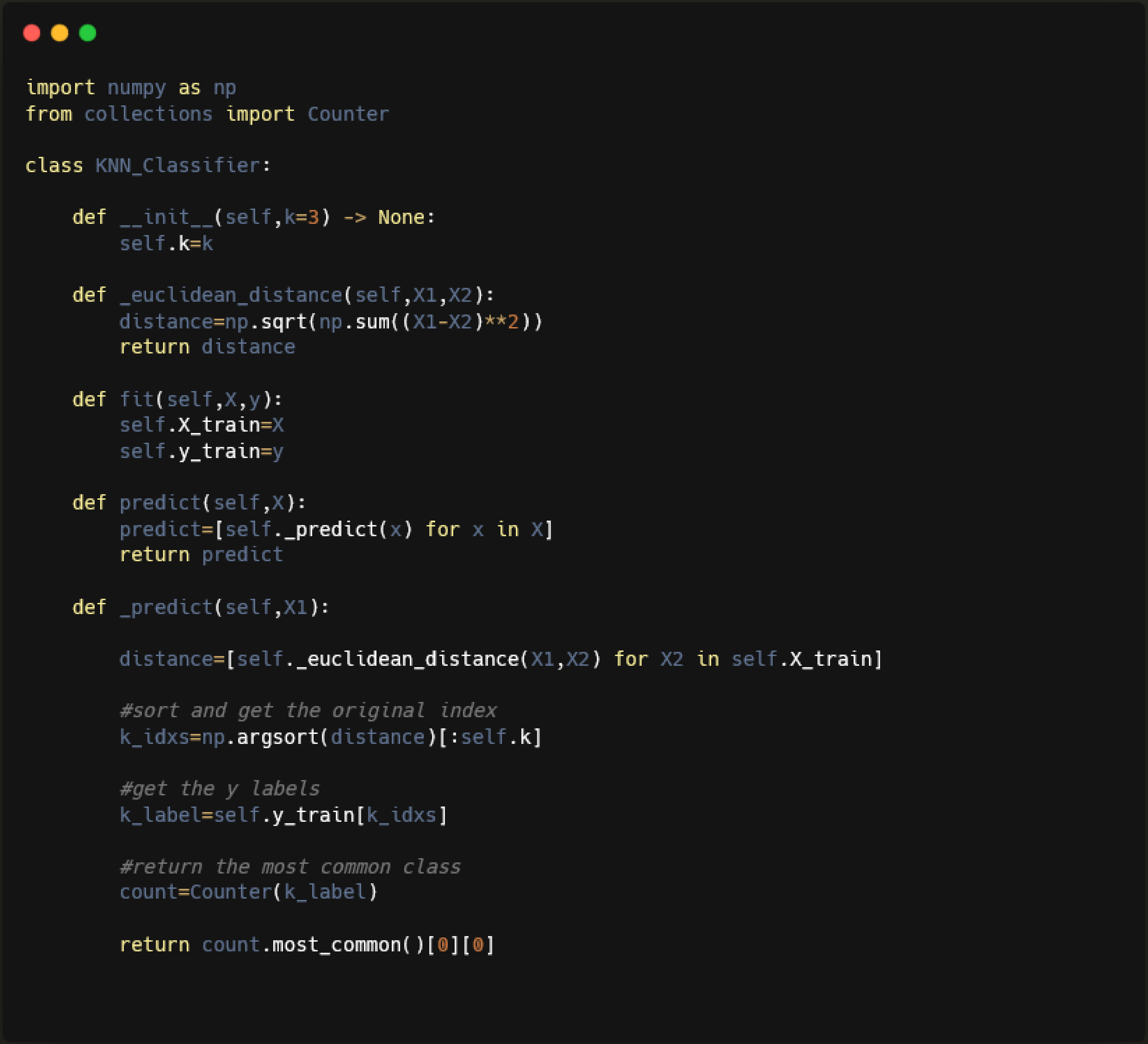
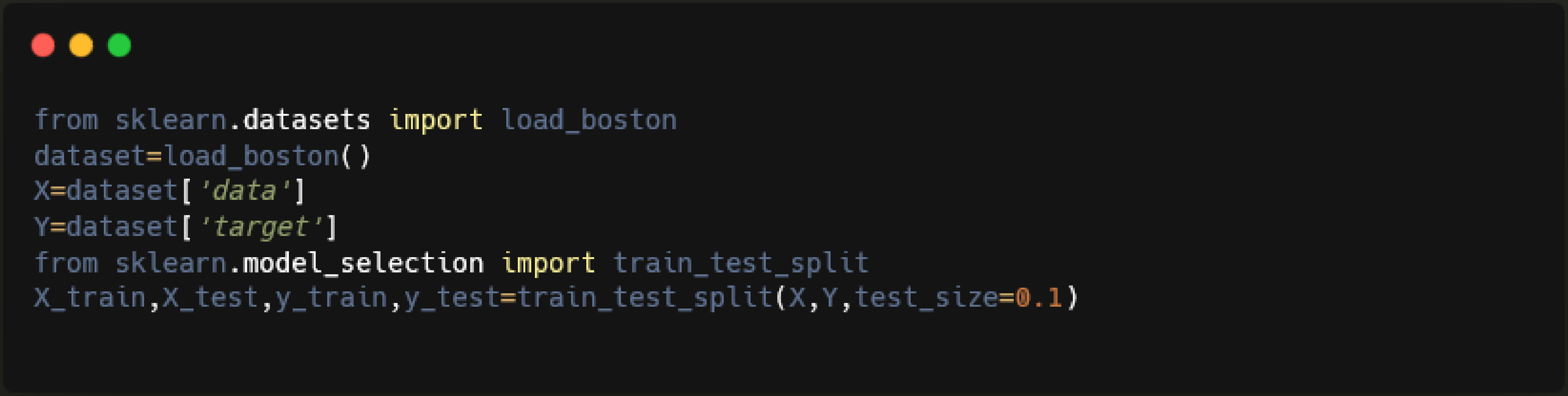
# K-NEAREST NEIGHBOR(KNN) FROM SCRATCH



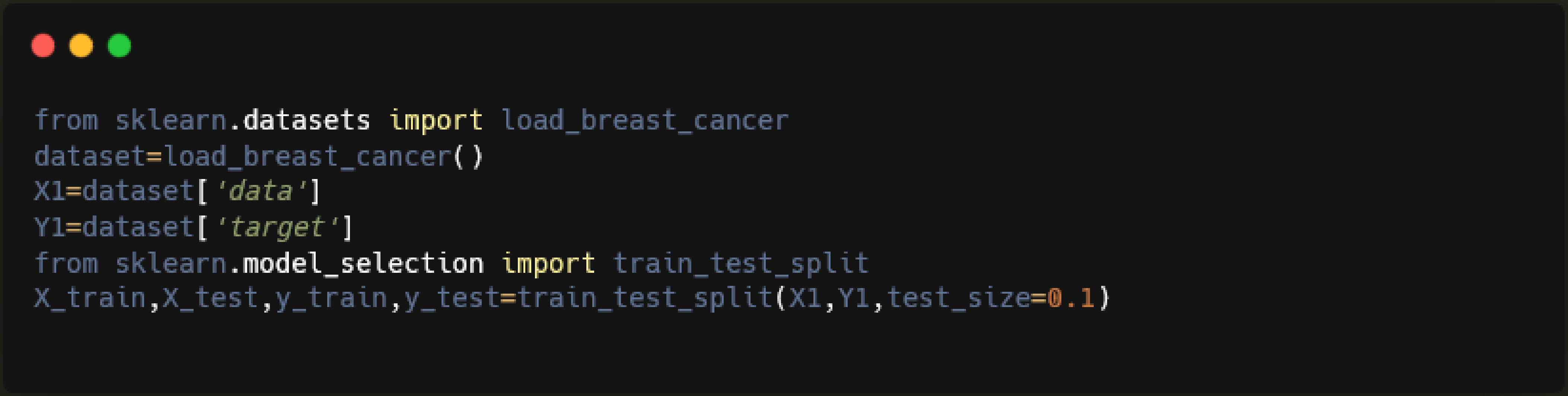
```
import numpy as np
from collections import Counter

class KNN_Regressor:

    def __init__(self,k=3) -> None:
        self.k=k
        #number of nearest neighbors

    def _ecludean_distance(self,X1,X2):
        return np.sqrt(np.sum((X1-X2)**2))

    def _predict(self,X1):

        distance=[self._ecludean_distance(X1,X2) for X2 in self.X_train]

        #sort and get the original index
        k_idxs=np.argsort(distance)[:self.k]


        k_labels=self.y_train[k_idxs]

        return np.mean(k_labels)

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

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



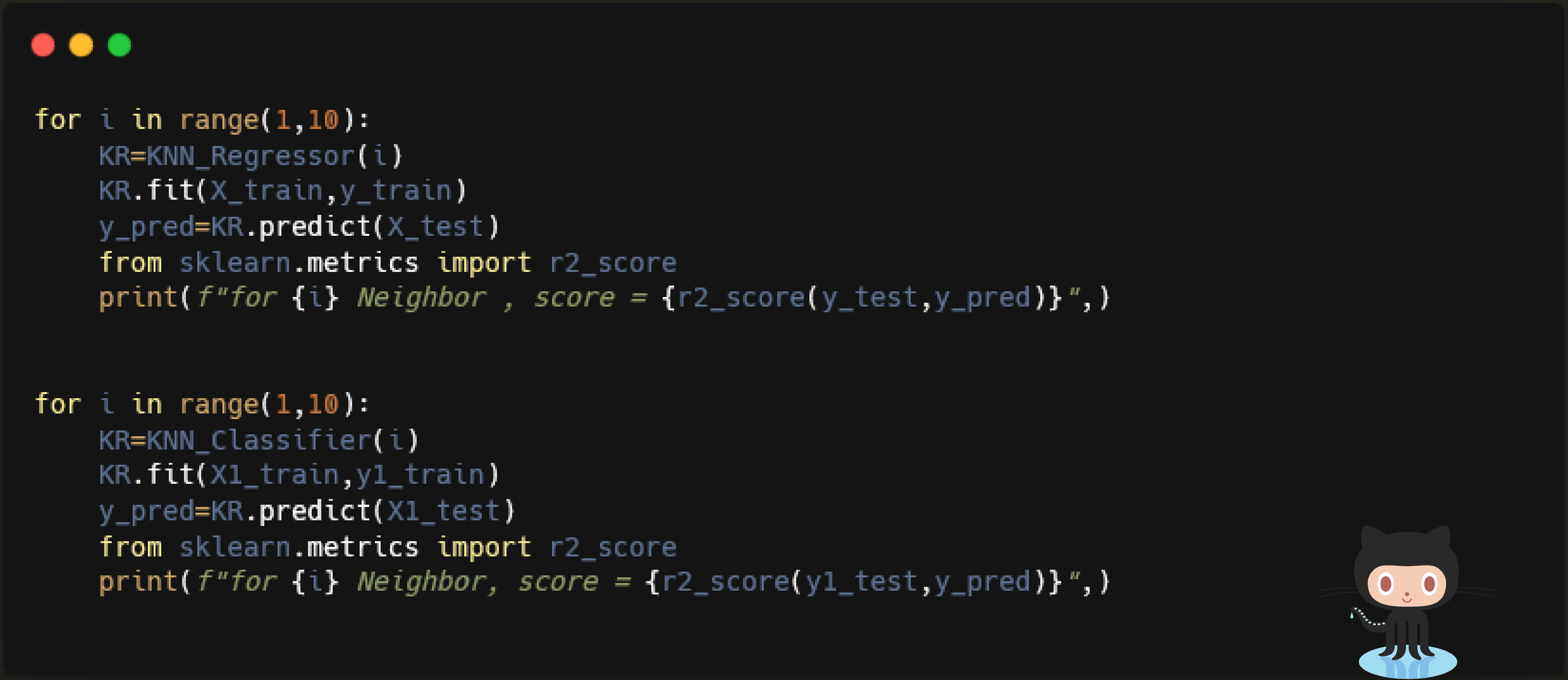
```python
import numpy as np
from collections import Counter

class KNN_Classifier:

    def __init__(self,k=3) -> None:
        self.k=k

    def _euclidean_distance(self,X1,X2):
        distance=np.sqrt(np.sum((X1-X2)**2))
        return distance

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

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

    def _predict(self,X1):

        distance=[self._euclidean_distance(X1,X2) for X2 in self.X_train]

        #sort and get the original index
        k_idxs=np.argsort(distance)[:self.k]

        #get the y labels
        k_label=self.y_train[k_idxs]

        #return the most common class
        count=Counter(k_label)

        return count.most_common()[0][0]
```



```
from sklearn.datasets import load_boston
dataset=load_boston()
X=dataset['data']
Y=dataset['target']
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.1)
```



```
from sklearn.datasets import load_breast_cancer
dataset=load_breast_cancer()
X1=dataset[‘data’]
Y1=dataset[‘target’]
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X1,Y1,test_size=0.1)
```



```
for i in range(1,10):
    KR=KNN_Regressor(i)
    KR.fit(X_train,y_train)
    y_pred=KR.predict(X_test)
    from sklearn.metrics import r2_score
    print(f"for {i} Neighbor , score = {r2_score(y_test,y_pred)}",)


for i in range(1,10):
    KR=KNN_Classifier(i)
    KR.fit(X1_train,y1_train)
    y_pred=KR.predict(X1_test)
    from sklearn.metrics import r2_score
    print(f"for {i} Neighbor, score = {r2_score(y1_test,y_pred)}",)
```

Click for Github

LINK