

```

1 # Pseudo code for k-Nearest Neighbors Surrogate model
2
3 # No guarantee of executability.
4
5 #####
6
7 # Training of the main and surrogate models with the case data
8
9 # Main model
10
11 main_model.fit(X_train, y_train)
12
13 # Surrogate model
14
15 surrogate_model.fit(X_train, y_train)
16
17 #####
18
19 # Prediction and explanation by the main and surrogate models with user's input
20
21 max_n_neighbors = 10
22
23 # Main model
24
25 y_pred = main_model.predict(X_in)
26
27 # Surrogate model
28
29 ndcg_k2_list = list()
30
31 for j in range(max_n_neighbors):
32     k = j + 1
33     y_pred_k = surrogate_model.predict(X_in, k)
34     ndcg_k2 = ndcg(y_pred, y_pred_k)
35     ndcg_k2_list.append(ndcg_k2)
36
37 k_closest = argmax(ndcg_k2_list) + 1
38 ndcg_k2_closest = ndcg_k2_list[k_closest - 1]
39
40 (neigh_dist, neigh_ind) = surrogate_model.kneighbors(X_in, k_closest)
41
42 meta_data_list = surrogate_model.search_medata(neigh_ind)
43
44 #####
45
46 # Evaluation of the main and surrogate models with the case data
47
48 n_tests = 6000
49 max_n_neighbors = 10
50
51 for i in range(n_tests):
52     X_in = X_test[i]
53     y_true = y_test[i]
54
55     # Main model
56
57     ndcg_nn = ndcg(y_true, y_pred)
58
59     # Surrogate model
60
61     ndcg_k1_list = list()
62     ndcg_k2_list = list()
63     for j in range(0, max_n_neighbors):
64         k = j + 1
65         y_pred_k = surrogate_model.predict(X_in, k)
66         ndcg_k1 = ndcg(y_true, y_pred_k)
67         ndcg_k2 = ndcg(y_pred, y_pred_k)
68         ndcg_k1_list.append(ndcg_k1)
69         ndcg_k2_list.append(ndcg_k2)
70
71     k_closest = argmax(ndcg_k2_list) + 1
72     ndcg_k1_closest = ndcg_k1_list[k_closest - 1]
73     ndcg_k2_closest = ndcg_k2_list[k_closest - 1]

```