

```

1 import numpy as np
2 import sklearn.svm as svm
3 def c_selection(x_train, y_train, x_validation, y_validation, kernel_types , debugging=False):
4     """
5     Find out which value of the hyper parameter 'C' is best for a particular C Support Vector Machine
6     This method will try random values 0 exclusive to max_iter inclusive, it will then find out some of the best
7     values of c and will then return a dictionary with the kernel type as the key, and the best 'c' value for that kernel
8     """
9
10    kernel_c_dict = {}
11    for kernel_type in kernel_types:
12        print(kernel_type + " evaluation:")
13        max_iter = 100
14        best_c_dict = {}
15        for iteration in range(max_iter):
16            svm_c_error = []
17            c_range = np.array((np.random.rand(20, )))
18            if iteration == 0:
19                c_range += iteration + 0.000001
20            else:
21                c_range += iteration
22
23            for c_value in c_range:
24                model = svm.SVC(kernel=kernel_type, C=c_value)
25                model.fit(X=x_train,y=y_train.ravel()) # DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y
26                error = 1. - model.score(x_validation, y_validation)
27                svm_c_error.append(error)
28
29            if debugging:
30                for C, error in zip(c_range, svm_c_error):
31                    print("C={}, error: {}".format(C, error))
32
33            ...
34            # Note: do we really need to determine all 'c' values with the same smallest error? no
35            searchval = min(svm_c_error)
36            best_c_indices = np.where(svm_c_error == searchval)[0] # note that svm_c_error is an type:<List> not a type<nd.array>
37            print(best_c_indices)
38            print("Best 'C' Hyper-Parameters:")
39            for index in best_c_indices:
40                print("\'C\': {}, with the error of {}".format(c_range[index], svm_c_error[index]))
41                best_c_dict[c_range[index]] = svm_c_error[index]
42            ...
43            c_index_with_least_error = svm_c_error.index(min(svm_c_error))
44            best_c_dict[c_range[c_index_with_least_error]] = svm_c_error[c_index_with_least_error]
45            best_c_value = min(best_c_dict.values())
46            best_c = list(best_c_dict.keys())[list(best_c_dict.values()).index(best_c_value)]
47            kernel_c_dict[kernel_type] = (best_c, best_c_value)
48
49    return kernel_c_dict

```