

use svc to solve a multiclass classification problem using one vs one and one vs rest approach also find the suitable approach for the problem

```
In [39]: 1 from sklearn.datasets import load_iris
2 from sklearn.svm import SVC
3 from sklearn.metrics import accuracy_score
4 from sklearn.model_selection import train_test_split
5 from sklearn.multiclass import OneVsOneClassifier, OneVsRestClassifier
```

```
In [40]: 1 iris = load_iris()
```

```
In [41]: 1 X,y = iris.data,iris.target
```

```
In [42]: 1 xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size=0.2, random_state=1)
```

```
In [64]: 1 ovr_model= OneVsRestClassifier(SVC())
2 ovr_model.fit(xtrain,ytrain)
3 ypred_ovr=ovr_model.predict(xtest)
4 ovr_accuracy=accuracy_score(ytest,ypred_ovr)
5 ovr_accuracy
```

```
Out[64]: 0.9666666666666667
```

```
In [65]: 1 ovo_model= OneVsOneClassifier(SVC())
2 ovo_model.fit(xtrain,ytrain)
3 ypred_ovo=ovo_model.predict(xtest)
4 ovo_accuracy=accuracy_score(ytest,ypred_ovo)
5 ovo_accuracy
```

```
Out[65]: 0.9666666666666667
```

```
In [66]: 1 print('Accuracy score for one vs one classifer is ',ovo_accuracy)
```

```
Accuracy score for one vs one classifer is  0.9666666666666667
```

```
In [67]: 1 print('Accuracy score for one vs one classifer is ',ovr_accuracy)
```

```
Accuracy score for one vs one classifer is  0.9666666666666667
```

```
In [68]: 1 if ovr_accuracy>ovo_accuracy:
2     print('one vs rest is suitable')
3 else:
4     print('one vs one is suitable')
```

```
one vs one is suitable
```

```
In [ ]: 1
```

```
In [ ]: 1
```

