

perform binary classification on diabetics dataset using svc, apply svc with linear kernel , svc with poly kernel by using cross validation to find best value for hyperparameter degree, apply radial basis kernel use cross validation to find the best value for gamma

```
In [1]: 1 import pandas as pd
        2 from sklearn.svm import SVC
        3 from sklearn.metrics import accuracy_score
        4 from sklearn.model_selection import train_test_split
```

```
In [3]: 1 data = pd.read_csv("pima.csv" , header = None)
        2 data.sample(5)
```

```
Out[3]:
```

	0	1	2	3	4	5	6	7	8
590	11	111	84	40	0	46.8	0.925	45	1
389	3	100	68	23	81	31.6	0.949	28	0
324	2	112	75	32	0	35.7	0.148	21	0
557	8	110	76	0	0	27.8	0.237	58	0
483	0	84	82	31	125	38.2	0.233	23	0

```
In [7]: 1 x=data.iloc[:, :-1]
        2 x.shape
```

```
Out[7]: (768, 8)
```

```
In [9]: 1 y=data.iloc[:, -1]
        2 y.shape
```

```
Out[9]: (768,)
```

```
In [16]: 1 xtrain,xtest,ytrain,ytest = train_test_split(x,y, test_size=0.2, random
        2 xtrain.shape,xtest.shape,ytrain.shape,ytest.shape)
```

```
Out[16]: ((614, 8), (154, 8), (614,), (154,))
```

```
In [66]: model_svc = SVC()
        model_svc.fit(xtrain,ytrain)
        ypred=model_svc.predict(xtest)
        accuracy=accuracy_score(ytest,ypred)
        print('linear kernel accuracy is ', accuracy)
```

```
linear kernel accuracy is  0.7662337662337663
```

```
In [71]: 1 model_poly = SVC(kernel='poly')
2 model_poly.fit(xtrain,ytrain)
3 ypred2=model_poly.predict(xtest)
4 accuracy1=accuracy_score(ytest,ypred2)
5 print('poly kernel accuracy before fine tuning is ', accuracy1)
```

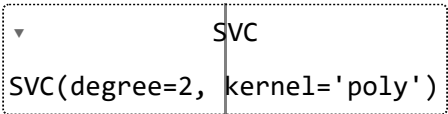
poly kernel accuracy before fine tuning is 0.7597402597402597

```
In [72]: 1 from sklearn.model_selection import GridSearchCV
```

```
In [107]: 1
2 from sklearn.model_selection import RepeatedKFold
3 cv=RepeatedKFold(n_splits=10,n_repeats=3, random_state=3)
4 params={'degree':[1,2,3,4]}
5
6 model2_poly=SVC(kernel='poly')
7 search = GridSearchCV(model2_poly,params,cv=cv)
8 result=search.fit(x,y)
9 result.best_params_
10
```

Out[107]: {'degree': 2}

```
In [108]: 1 model3_poly = SVC(kernel="poly", degree=2)
2 model3_poly.fit(xtrain,ytrain)
```

Out[108]: 

```
In [109]: 1 ypred3=model3_poly.predict(xtest)
2 accuracy3=accuracy_score(ytest,ypred3)
3 print('poly kernel after fine tuning accuracy is ', accuracy3)
```

poly kernel after fine tuning accuracy is 0.7532467532467533

```
In [110]: 1 model_rbf = SVC(kernel='rbf')
2 model_rbf.fit(xtrain,ytrain)
3 ypred4=model_rbf.predict(xtest)
4 accuracy4=accuracy_score(ytest,ypred4)
5 print('rbf kernel accuracy before fine tuning is ', accuracy4)
```

rbf kernel accuracy before fine tuning is 0.7662337662337663

```
In [111]: 1 from sklearn.model_selection import RepeatedKFold
2 cv=RepeatedKFold(n_splits=10,n_repeats=3, random_state=6)
3 params={'gamma':[0.1, 0.02,1,4]}
4
5 model2_rbf=SVC(kernel='rbf')
6 search = GridSearchCV(model2_rbf,params,cv=cv)
7 result=search.fit(x,y)
8 result.best_params_
9
```

Out[111]: {'gamma': 0.02}

```
In [112]: 1 model3_rbf = SVC(kernel='rbf', gamma=2)
          2 model3_rbf.fit(xtrain,ytrain)
```

```
Out[112]: SVC
          SVC(gamma=2)
```

```
In [113]: 1 ypred5=model3_rbf.predict(xtest)
          2 accuracy5=accuracy_score(ytest,ypred5)
          3 print('poly kernel accuracy after fine tuning is ', accuracy5)
```

poly kernel accuracy after fine tuning is 0.6428571428571429

```
In [ ]: 1
```

```
In [ ]: 1
```