In [3]:
```
!pip install scikit-learn
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: scikit-learn in c:\users\arooj\appdata\roaming\python\python311\site-packages (1.3.2)
Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn)
(1.24.3)
Requirement already satisfied: scipy>=1.5.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.11.
1)
Requirement already satisfied: joblib>=1.1.1 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1.2.
0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-lear
n) (2.2.0)
```

In [6]:
```
import sklearn
# print(sklearn.__version__)
```

```
1.3.2
```

In [42]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn import svm                # its used for regression or classification
```

In [43]:
```
X, y = load_iris(return_X_y=True)   # loading the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

# Train size

0.8 mean 80% of data means ration 80 , 20

In [71]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=0)
```

example two here chng train zise and syntax must follow this

In [72]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.6, random_state=0)
```

In [73]:
```
print(X.shape)
```

```
(150, 4)
```

In [74]:
```
print (y.shape)
```

```
(150,)
```

In [75]:
```
print(X_train.shape)
```

```
(90, 4)
```

In [76]:
```
print(y_train.shape)
```

```
(90,)
```

In [77]:
```
clf_svm = svm.SVC()                    # SVC for classification  >>> initilize the classifier >>> algorithm
```

In [78]:
```
clf_svm.fit(X_train, y_train)                    #  FIRST TRAIN THEN TESTING >> x data >> y for lable
                                                  # giving classifier training
```

Out[78]:
```
SVC()
```

In [79]:
```
svm_acc=clf_svm.score(X_test , y_test)                    # testing using svm
```

In [80]:
```
print(svm_acc)
```

```
0.9333333333333333
```

In [81]:
```
print(svm_acc* 100)
```

```
93.33333333333333
```

In [82]:
```
svm_acc_test=clf_svm.score(X_test , y_test)          # testing set accc using svm
```

In [84]:
```
svm_acc_train=clf_svm.score(X_train , y_train)       # trainig set accuracy using svm
```

In [92]:
```
print("*********** testing accuracy ***********")
print(svm_acc_test* 100)
```

```
*********** testing accuracy ***********
93.33333333333333
```

In [93]:
```
print("*********** traning accuracy ***********")
print(svm_acc_train* 100)
```

```
*********** traning accuracy ***********
97.77777777777777
```

In [95]:
```
from sklearn.model_selection import cross_val_score   # for cross validation >> no of instances according to class
```

In [103…
```
 cv_svm = cross_val_score(clf_svm, X,y, cv=5)       # 5 ka cross validation show hoga >>> only for testing accuracies
print (cv_svm)
```

```
[0.96666667 0.96666667 0.96666667 0.93333333 1.        ]
```

In [108…
```
print(round(svm_acc_train,2)*100)
```

```
98.0
```

In [104…
```
# for mean
print(cv_svm.mean())                        # mean accuracy and standard deviation
```

```
0.9666666666666666
```

In [105…
```
print(cv_svm.std())
```

```
0.02108185106778919
```

# PPr mn questions yeh aye thy

In [106…
```
# 3 different train test splits >>>>> traning and testing accuracies >>>>>>>>> dataset bhi given hoga
```

In [107…
```
# cross validation ki three different splits 3,4>>> 5,4 mn standard deviation baatani ha code kr ke
```

# Assignment

In [110…
```
# table do datasets pe kamm krna training or testing accuracy check and also check cross validation overall mean accu
```

In [ ]:
```

```