SVC*

```
from sklearn.datasets import load_iris
iris= load_iris()
dir(iris)
```

```
['DESCR',
 'data',
 'data_module',
 'feature_names',
 'filename',
 'frame',
 'target',
 'target_names']
```

```
print(iris.feature_names)
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

```
print(iris.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

```
import pandas as pd
df=pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
target=pd.DataFrame(iris.target,columns=['target'])
dataset=pd.concat([df,target],axis='columns')
dataset
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

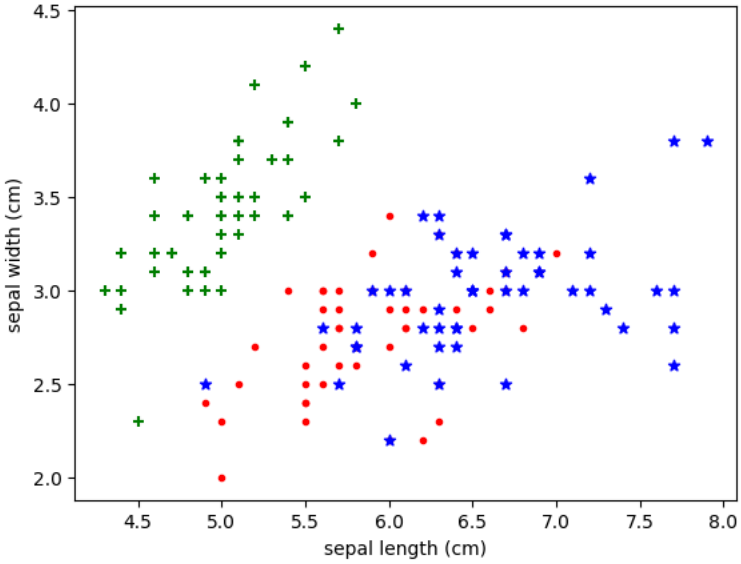
```
from matplotlib import pyplot as plt
%matplotlib inline
```

```
df0=dataset[dataset.target==0]
df1=dataset[dataset.target==1]
df2=dataset[dataset.target==2]
df0.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

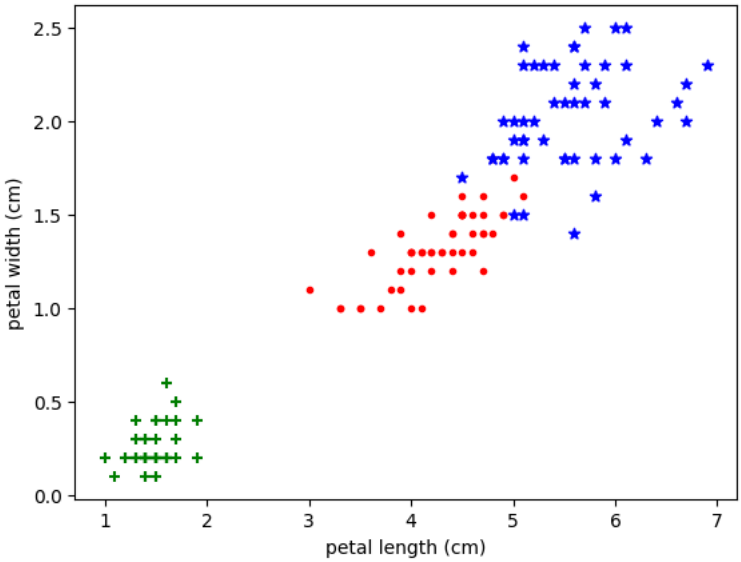
```
plt.xlabel('sepal length (cm)')
plt.ylabel('sepal width (cm)')
plt.scatter(df0['sepal length (cm)'],df0['sepal width (cm)'],color='green',marker='+')
plt.scatter(df1['sepal length (cm)'],df1['sepal width (cm)'],color='red',marker='.')
plt.scatter(df2['sepal length (cm)'],df2['sepal width (cm)'],color='blue',marker='*')
```

```
<matplotlib.collections.PathCollection at 0x7eed6d8bd5d0>
```




```
plt.xlabel('petal length (cm)')
plt.ylabel('petal width (cm)')
plt.scatter(df0['petal length (cm)'],df0['petal width (cm)'],color='green',marker='+')
plt.scatter(df1['petal length (cm)'],df1['petal width (cm)'],color='red',marker='.')
plt.scatter(df2['petal length (cm)'],df2['petal width (cm)'],color='blue',marker='*')
```

```
<matplotlib.collections.PathCollection at 0x7eeda670c820>
```



The SVM will be able to perform better as it can draw a very nice and clear boundary between df0 and df1

```
from sklearn.model_selection import train_test_split
X=dataset.drop(['target'],axis='columns')
y=dataset.target
X_train,X_test,y_train,y_test=train_test_split(X,y,random_state=44,test_size=0.2)
```

```
from sklearn.svm import SVC
model=SVC()
model.fit(X_train,y_train)
```

```
SVC()
```

```
model.score(X_test,y_test)
```

```
0.9666666666666667
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
model_rf=RandomForestClassifier(n_estimators=10)
model_rf.fit(X_train,y_train)
```

```
RandomForestClassifier(n_estimators=10)
```

```
model.score(X_test,y_test)
```

```
0.9666666666666667
```

KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train,y_train)
```

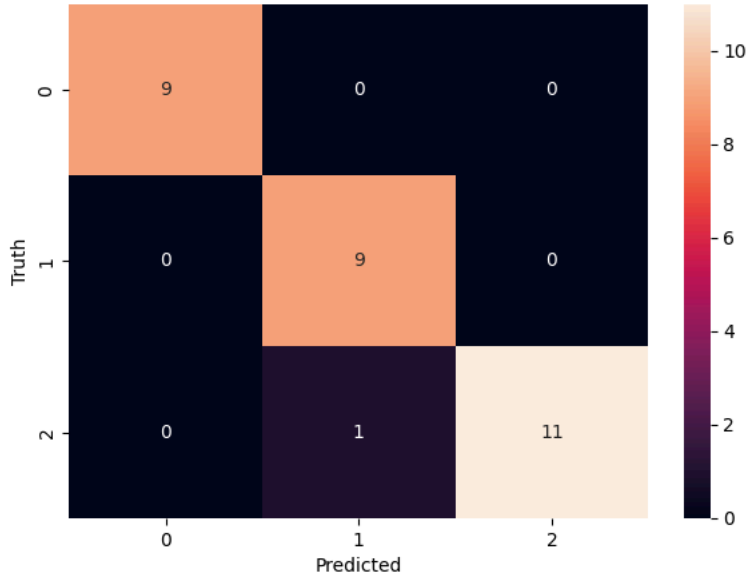
```
KNeighborsClassifier(n_neighbors=3)
```

```
knn.score(X_test,y_test)
```

```
0.9666666666666667
```

```
from sklearn.metrics import confusion_matrix
y_pred=knn.predict(X_test)
cm=confusion_matrix(y_test,y_pred)
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(7,5))
sn.heatmap(cm,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(58.222222222222214, 0.5, 'Truth')
```



```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.90      1.00      0.95         9
           2       1.00      0.92      0.96        12

    accuracy                           0.97        30
   macro avg       0.97      0.97      0.97        30
weighted avg       0.97      0.97      0.97        30
```