

## 2. Use the data set "fruits.csv" and form SVM classification by taking weight and size as the parameter. If you take single parameters does the SVM technique differs? Justify.

```
In [6]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import metrics
df = pd.read_csv(r'C:\Users\Nirmalya Majhi\Desktop\Advanced IT Workshop\fruits.csv')
print(df.head())
#For two parameters
X = df.drop(['Index', 'Class'], axis=1)
Y = df['Class']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=4)
classifier = SVC(kernel='rbf')
classifier.fit(X_train, Y_train)
Y_pred = classifier.predict(X_test)
print("\nFor 2 Parameters:")
print("Confusion Matrix: \n", metrics.confusion_matrix(Y_test, Y_pred))
print(metrics.classification_report(Y_test, Y_pred))
print("Accuracy = ", (metrics.accuracy_score(Y_pred, Y_test)*100), "%")
#For one parameter
X = df.drop(['Index', 'Class', 'size'], axis=1)
Y = df['Class']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=4)
classifier = SVC(kernel='rbf')
classifier.fit(X_train, Y_train)
Y_pred = classifier.predict(X_test)
print("\nFor 1 Parameter:")
print("Confusion Matrix: \n", metrics.confusion_matrix(Y_test, Y_pred))
print(metrics.classification_report(Y_test, Y_pred))
print("Accuracy = ", (metrics.accuracy_score(Y_pred, Y_test)*100), "%")
```

	Index	Weight	size	Class
0	0	69	4.39	orange
1	1	68	4.21	orange
2	2	65	4.28	orange
3	3	72	5.85	apple
4	4	67	4.74	orange

For 2 Parameters:

Confusion Matrix:

```
[[2 0]
 [0 2]]
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	2
orange	1.00	1.00	1.00	2
accuracy			1.00	4
macro avg	1.00	1.00	1.00	4
weighted avg	1.00	1.00	1.00	4

Accuracy = 100.0 %

For 1 Parameter:

Confusion Matrix:

```
[[1 1]
 [0 2]]
```

	precision	recall	f1-score	support
apple	1.00	0.50	0.67	2
orange	0.67	1.00	0.80	2
accuracy			0.75	4
macro avg	0.83	0.75	0.73	4
weighted avg	0.83	0.75	0.73	4

Accuracy = 75.0 %

Discussion: With single parameter, the accuracy of the model decreases. Thus it is required that we use more number of useful attributes or parameters to train our model more effectively.