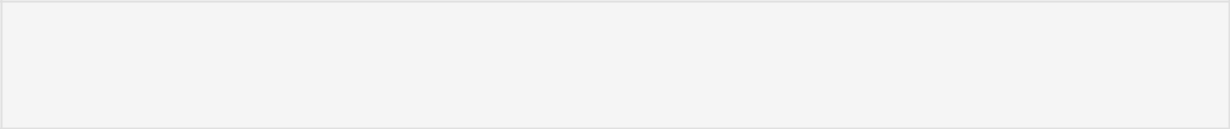
LabAssignment\_3 07/12/22, 11:08 PM



In [1]:

**import** pandas **as** pd

data **=** pd**.**read\_csv("apples\_and\_oranges.csv")

data**.**head()

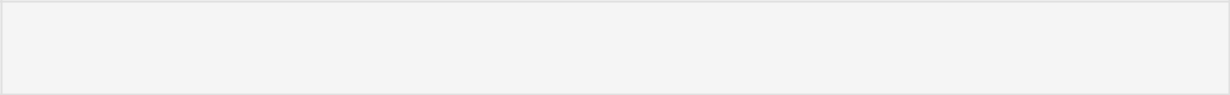
Out[1]: Weight Size Class



1. 69 4.39 orange
2. 69 4.21 orange
3. 65 4.09 orange



|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 72 | 5.85 | apple |
|  |  |  |  |
| 4 | 67 | 4.70 | orange |
|  |  |  |  |

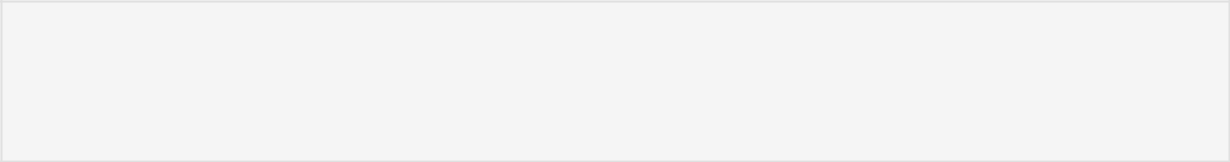


In [2]:

In [3]:

**from** sklearn.model\_selection **import** train\_test\_split

training\_set, test\_set **=** train\_test\_split(data, test\_size **=** 0.2, random\_s

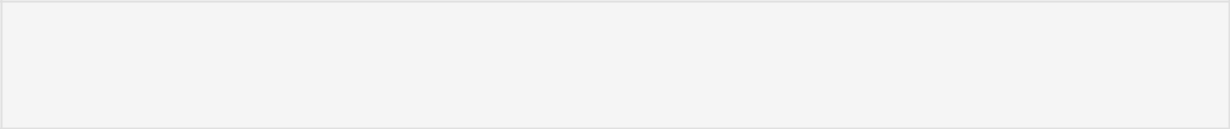


X\_train **=** training\_set**.**iloc[:,0:2]**.**values

Y\_train **=** training\_set**.**iloc[:,2]**.**values

X\_test **=** test\_set**.**iloc[:,0:2]**.**values

Y\_test **=** test\_set**.**iloc[:,2]**.**values



In [4]:

**from** sklearn.svm **import** SVC

classifier **=** SVC(kernel**=**'rbf', random\_state **=** 1)

classifier**.**fit(X\_train,Y\_train)

Out[4]:

In [5]:

In [6]:

In [7]:

Out[7]:

SVC(random\_state=1)



Y\_pred **=** classifier**.**predict(X\_test)



test\_set["Predictions"] **=** Y\_pred



test\_set

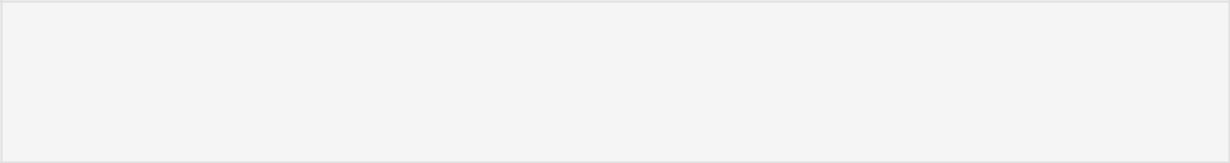
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Weight | Size | Class | Predictions |
|  |  |  |  |  |
| 2 | 65 | 4.09 | orange | apple |
|  |  |  |  |  |
| 31 | 66 | 4.68 | orange | apple |
|  |  |  |  |  |
| 3 | 72 | 5.85 | apple | apple |
|  |  |  |  |  |
| 21 | 70 | 4.83 | orange | apple |
|  |  |  |  |  |
| 27 | 70 | 4.22 | orange | apple |
|  |  |  |  |  |
| 29 | 71 | 5.26 | apple | apple |
|  |  |  |  |  |
| 22 | 69 | 4.61 | orange | apple |
|  |  |  |  |  |
| 39 | 73 | 5.03 | apple | apple |
|  |  |  |  |  |

http://localhost:8888/nbconvert/html/Documents/MTECH(DSA)/MyLabWork/AML\_Lab/LabAssignment\_3.ipynb?download=false Page 1 of 3

LabAssignment\_3 07/12/22, 11:08 PM

In [8]:

Calculating the accuracy of the predictions



**from** sklearn.metrics **import** confusion\_matrix cm **=** confusion\_matrix(Y\_test,Y\_pred)

accuracy **=** float(cm**.**diagonal()**.**sum())**/**len(Y\_test)

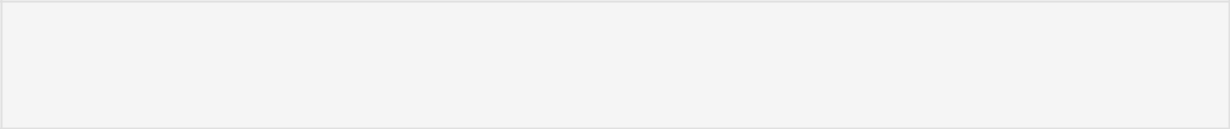
print("\nAccuracy Of SVM For The Given Dataset : ", accuracy)

Accuracy Of SVM For The Given Dataset : 0.375

Visualizing the classifier

In [9]:

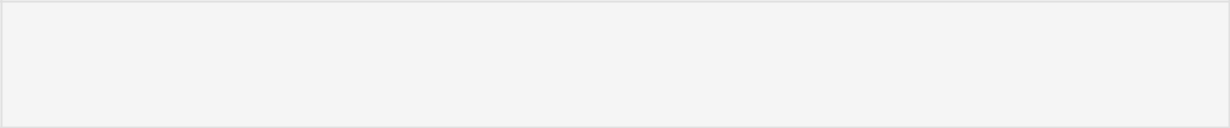
Before we visualize we might need to encode the classes ‘apple’ and ‘orange’ into numericals.We can achieve that using the label encoder.



**from** sklearn.preprocessing **import** LabelEncoder le **=** LabelEncoder()

Y\_train **=** le**.**fit\_transform(Y\_train)

After encoding , fit the encoded data to the SVM



In [10]:

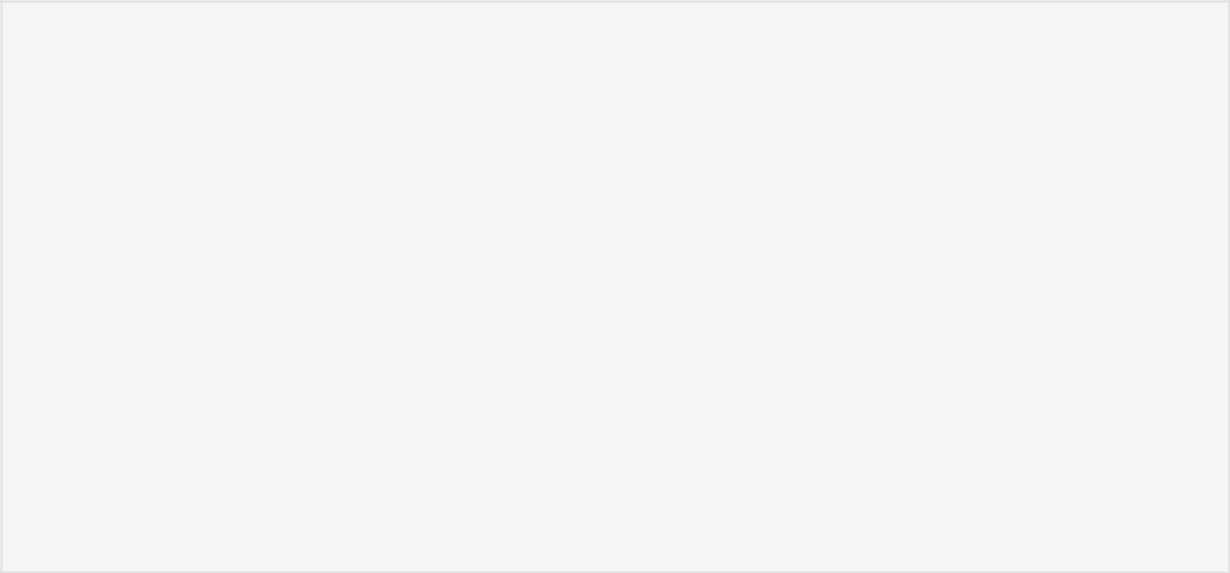
**from** sklearn.svm **import** SVC

classifier **=** SVC(kernel**=**'rbf', random\_state **=** 1)

classifier**.**fit(X\_train,Y\_train)

Out[10]: SVC(random\_state=1)

Let’s Visualize!



In [11]:

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**from** matplotlib.colors **import** ListedColormap plt**.**figure(figsize **=** (7,7))

X\_set, y\_set **=** X\_train, Y\_train

X1, X2 **=** np**.**meshgrid(np**.**arange(start **=** X\_set[:, 0]**.**min() **-** 1, stop **=** X\_se

plt**.**contourf(X1, X2, classifier**.**predict(np**.**array([X1**.**ravel(), X2**.**ravel()]

plt**.**xlim(X1**.**min(), X1**.**max())

plt**.**ylim(X2**.**min(), X2**.**max())

**for** i, j **in** enumerate(np**.**unique(y\_set)):

plt**.**scatter(X\_set[y\_set **==** j, 0], X\_set[y\_set **==** j, 1], c **=** ListedCol

plt**.**title('Apples Vs Oranges')

plt**.**xlabel('Weight In Grams')

plt**.**ylabel('Size in cm')

plt**.**legend()

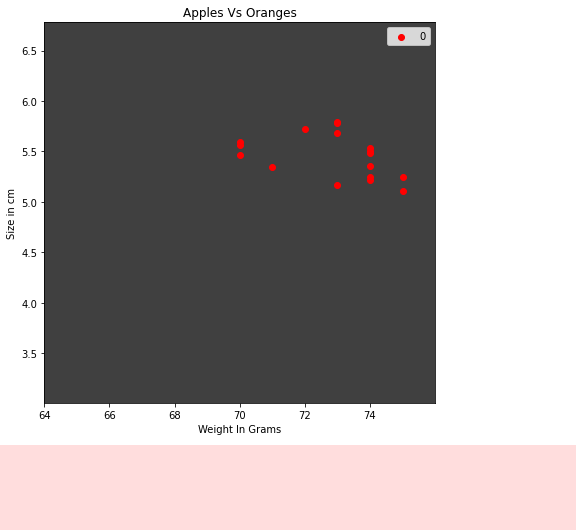
plt**.**show()



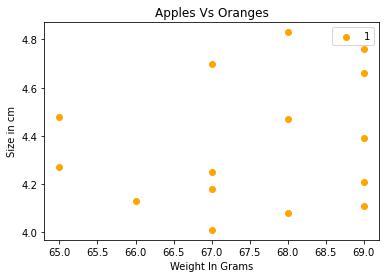
\*c\* argument looks like a single numeric RGB or RGBA sequence, which shou ld be avoided as value-mapping will have precedence in case its length ma tches with \*x\* & \*y\*. Please use the \*color\* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGB A value for all points.

http://localhost:8888/nbconvert/html/Documents/MTECH(DSA)/MyLabWork/AML\_Lab/LabAssignment\_3.ipynb?download=false Page 2 of 3

LabAssignment\_3 07/12/22, 11:08 PM



\*c\* argument looks like a single numeric RGB or RGBA sequence, which shou ld be avoided as value-mapping will have precedence in case its length ma tches with \*x\* & \*y\*. Please use the \*color\* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGB A value for all points.



http://localhost:8888/nbconvert/html/Documents/MTECH(DSA)/MyLabWork/AML\_Lab/LabAssignment\_3.ipynb?download=false Page 3 of 3