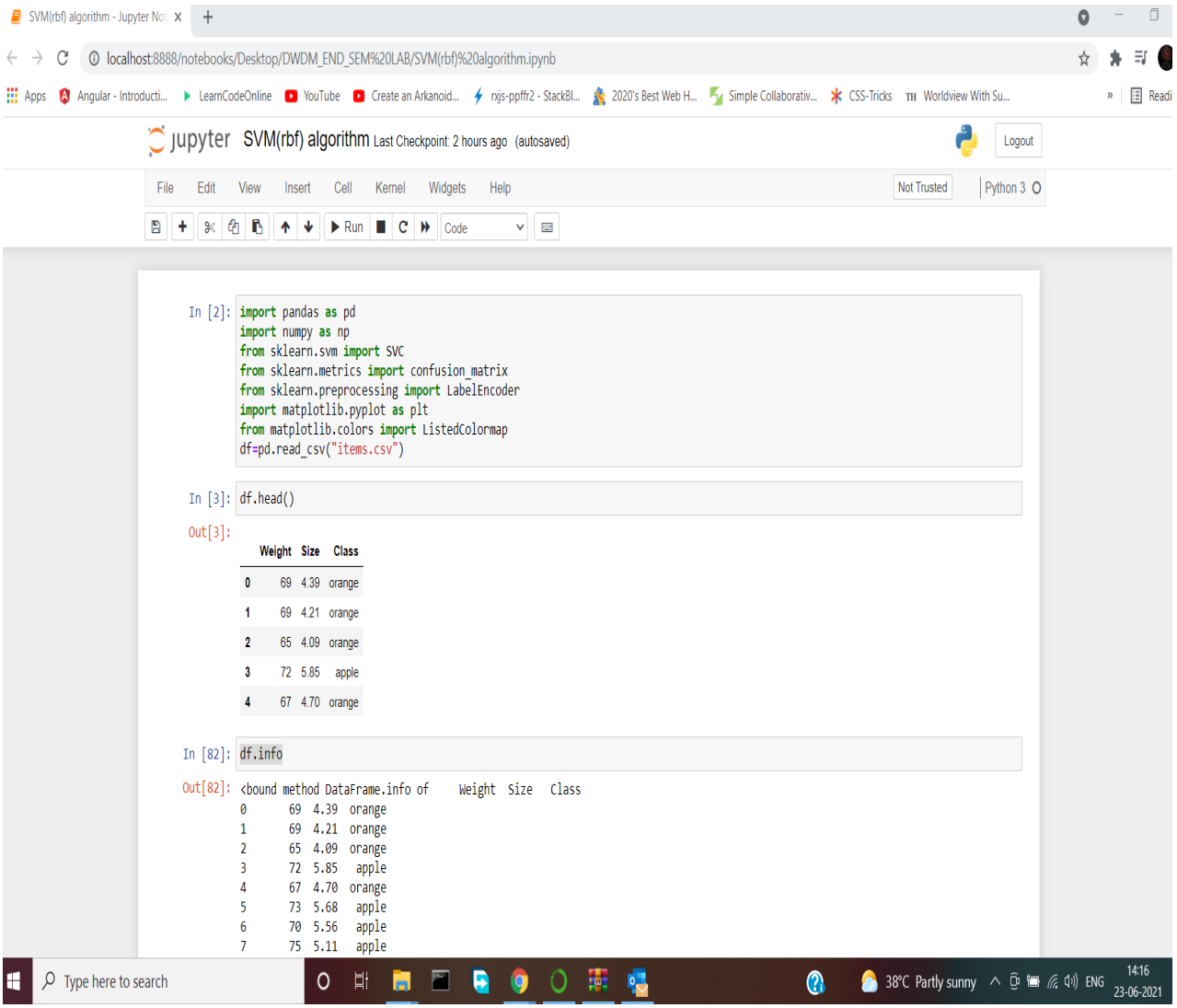
set (3) DWDM END LAB -SVM(rbf) algorithm

18BCS037

GANESH SETHU

1.items.csv dataset choosen

Kernel Type = Radial basis function (RBF), gamma=0.8



```
from sklearn.model_selection import train_test_split
```

```
training_set, test_set = train_test_split(df, test_size = 0.2, random_state = 1)
```

```
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
```

```
classifier = SVC(kernel='rbf', random_state = 1, gamma=0.6)
classifier.fit(X_train,Y_train)
```

```
SVC(gamma=0.6, random_state=1)
```

```
Y_prediction = classifier.predict(X_test)
```

In [11]:
```
test_set
```

Out[11]:

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

In [12]:
```
cm = confusion_matrix(Y_test,Y_prediction)
accuracy = float(cm.diagonal().sum())/len(Y_test)
print("Accuracy Of SVM For The Given Dataset : ", accuracy)
```

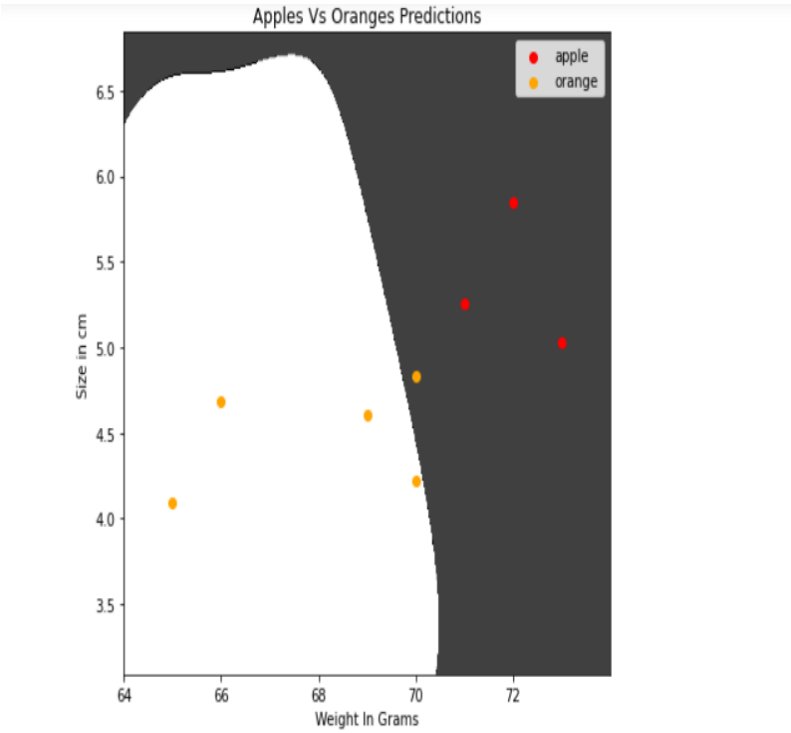
```
Accuracy Of SVM For The Given Dataset :  0.875
```

In [13]:
```
le = LabelEncoder()
Y_train = le.fit_transform(Y_train)
```

In [23]:
```
classifier = SVC(kernel='rbf', random_state = 1, gamma=0.8)
classifier.fit(X_train,Y_train)
```

Out[23]:
```
SVC(gamma=0.8, random_state=1)
```

In [25]:
```
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_set[:, 0].max() + 1, step = 0.01), np.arange(start = X_set
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape), alpha = 0.75, cmap = ListedColor
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 = ListedColormap(('red', 'orange'))(i), label = j)
plt.title('Apples Vs Oranges')
plt.xlabel('Weight In Grams')
plt.ylabel('Size in cm')
plt.legend()
plt.show()
```



```
# Predicted Output
```

```
plt.figure(figsize = (7,7))
X_set, y_set = X_test, Y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),np.arange(start = X_se
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),alpha = 0.75, cmap = ListedColo
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 = ListedColormap(('red', 'orange'))(i), label = j)
plt.title('Apples Vs Oranges Predictions')
plt.xlabel('Weight In Grams')
plt.ylabel('Size in cm')
plt.legend()
plt.show()
```

In [86]:
```
Y_pred = classifier.predict(X_test)
```

In [87]:
```
test_set["Predictions"] = Y_pred
```

```
<ipython-input-87-946a65001e17>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
rsus-a-copy
  test_set["Predictions"] = Y_pred
```

In [79]:
```
test_set
```

Out[79]:

	Weight	Size	Class	Predictions
2	65	4.09	orange	1
31	66	4.68	orange	1
3	72	5.85	apple	0
21	70	4.83	orange	0
27	70	4.22	orange	1
29	71	5.26	apple	0
22	69	4.61	orange	1
39	73	5.03	apple	0