In [21]:
```
Aim: To implement ensemble technique Boosting
```

In [ ]:
```
import numpy as nm
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from matplotlib.colors import ListedColormap
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn import model_selection
from sklearn.ensemble import AdaBoostClassifier
import warnings
warnings.filterwarnings('ignore')
```

In [7]:
```
dataset = pd.read_csv('Logistic_Iris.csv')
```

In [8]:
```
x = dataset.iloc[:, [0,1,2,3]].values
y = dataset.iloc[:, 4].values
```

In [9]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.25, random_state=0)
```

In [11]:
```
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)
```

In [12]:
```
adaboost = AdaBoostClassifier(n_estimators = 50, learning_rate = 0.2)
adaboost. fit(xtrain, ytrain)
```

Out[12]:
```
AdaBoostClassifier

AdaBoostClassifier(learning_rate=0.2)
```

In [13]:
```
adaboost.score(xtest, ytest)
```

Out[13]:
```
0.8947368421052632
```

In [14]:
```
y_pred = adaboost.predict(xtest)
print("Predicted values for AdaBoost Classifier:")
y_pred
```

```
Predicted values for AdaBoost Classifier:
```

Out[14]:
```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica'], dtype=object)
```

In [16]:
```
Acc_adaboost= accuracy_score(ytest,y_pred)*100
print ("\n\nTest Accuracy using AdaBoost Classifier: ", Acc_adaboost)
```

```


Test Accuracy using AdaBoost Classifier:  89.47368421052632
```

In [18]:
```
cm = confusion_matrix(ytest, y_pred)
print ("\n\n Confusion Matrix for AdaBoost Classifier: \n", cm)
```

```


 Confusion Matrix for AdaBoost Classifier:
 [[13  0  0]
 [ 0 15  1]
 [ 0  3  6]]
```

In [20]:
```
fig, ax = plt.subplots(figsize=(6, 6))
ax.imshow(cm)
ax.grid(False)
ax.xaxis.set(ticks=(0,1,2), ticklabels=('Predicted Setosa', 'Predicted Versicolor', 'Predicted Virginica'))
ax.yaxis.set(ticks=(0,1,2), ticklabels=('Actual Setosa', 'Actual Versicolor', 'Actual Virginica'))
ax.set_ylim(2.5, -0.5)
for i in range(3):
    for j in range(3):
        ax.text(j, i, cm[i, j], ha='center', va='center', color='white')
plt.show()
```