In [ ]:
```
Aim: To implement k-nearest neighbor algorithm
```

In [17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix,accuracy_score
```

In [18]:
```
dataset = pd.read_csv('iris.csv')
```

In [19]:
```
x = dataset.iloc[:, [2,3]].values
```

In [20]:
```
x
```

Out[20]:
```
array([[1.4, 0.2],
       [1.4, 0.2],
       [1.3, 0.2],
       [1.5, 0.2],
       [1.4, 0.2],
       [1.7, 0.4],
       [1.4, 0.3],
       [1.5, 0.2],
       [1.4, 0.2],
       [1.5, 0.1],
       [1.5, 0.2],
       [1.6, 0.2],
       [1.4, 0.1],
       [1.1, 0.1],
       [1.2, 0.2],
       [1.5, 0.4],
       [1.3, 0.4],
       [1.4, 0.3],
       [1.7, 0.3],
       [1.5, 0.3],
```

In [21]:
```
y = dataset.iloc[:, 4].values
```

In [22]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [23]:
```
sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)
```

In [24]:
```
knn = KNeighborsClassifier(n_neighbors = 7)
```

In [25]:
```
knn.fit(xtrain,ytrain)
```

Out[25]:
```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=7)
```

In [26]:
```
KNeighborsClassifier(n_neighbors = 7)
```

Out[26]:
```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=7)
```

In [27]:
```
ypred = knn.predict(xtest)
print(ypred)
```

```
['Virginica' 'Versicolor' 'Setosa' 'Virginica' 'Setosa' 'Virginica'
 'Setosa' 'Versicolor' 'Versicolor' 'Versicolor' 'Versicolor' 'Versicolor'
 'Versicolor' 'Versicolor' 'Versicolor' 'Setosa' 'Versicolor' 'Versicolor'
 'Setosa' 'Setosa' 'Virginica' 'Versicolor' 'Setosa' 'Setosa' 'Virginica'
 'Setosa' 'Setosa' 'Versicolor' 'Versicolor' 'Setosa' 'Virginica'
 'Versicolor' 'Setosa' 'Virginica' 'Virginica' 'Versicolor' 'Setosa'
 'Versicolor']
```

In [28]:
```
knn.score(xtest,ytest)
```

Out[28]:
```
0.9736842105263158
```

In [29]:
```
knn.score(xtrain,ytrain)
```

Out[29]:
```
0.9553571428571429
```

In [30]:
```
cm = confusion_matrix(ytest, ypred)
print("confusion matrix: \n", cm)
```

```
confusion matrix:
 [[13  0  0]
 [ 0 16  0]
 [ 0  1  8]]
```

In [ ]:
```

```

In [31]:
```
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()
```