In [ ]:
```
Aim:to implement new base classification for both iris and car sales datasets
```

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix,accuracy_score
from sklearn.naive_bayes import GaussianNB
```

In [10]:
```
dataset=pd.read_csv("Logistic_Iris.csv")
dataset1=pd.read_csv("Logistic_car_data.csv")
```

In [11]:
```
x=dataset.iloc[:,[0,1,2,3]]
y=dataset.iloc[:,4]
x1=dataset1.iloc[:,[2,3]]
y1=dataset1.iloc[:,4]
```

In [12]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25,random_state=49)
x1train,x1test,y1train,y1test=train_test_split(x1,y1,test_size=0.25,random_state=49)
```

In [13]:
```
sc=StandardScaler()
xtrain=sc.fit_transform(xtrain)
xtest=sc.transform(xtest)
x1train=sc.fit_transform(x1train)
x1test=sc.transform(x1test)
```

In [14]:
```
classifier=GaussianNB()
classifier.fit(xtrain,ytrain)
classifier1=GaussianNB()
classifier1.fit(x1train,y1train)
```

Out[14]:
```
GaussianNB()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [15]:
```
y_pred=classifier.predict(xtest)
y_pred1=classifier1.predict(x1test)
```

In [16]:
```
print('predicted values:',y_pred)
```

```
predicted values: ['Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-setosa' 'Iris-virginica']
```

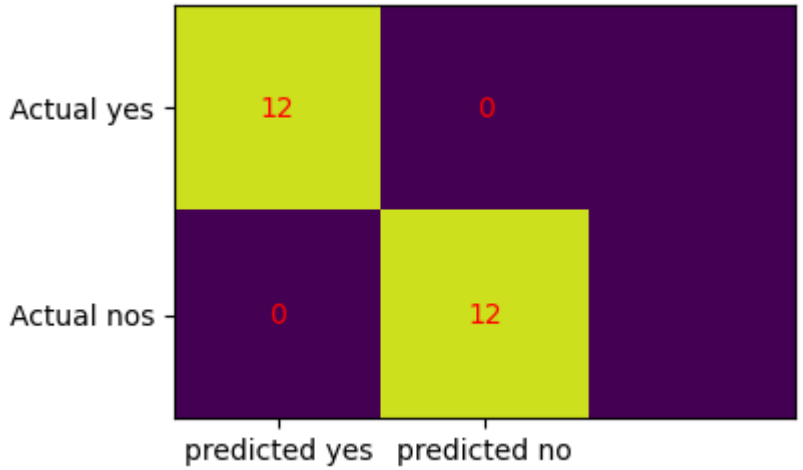
In [17]:
```
accuracy=accuracy_score(ytest,y_pred)*100
print("\n\nAccuracy using naive bayes:",accuracy)
accuracy1=accuracy_score(y1test,y_pred1)*100
print("\n\nAccuracy using naive bayes:",accuracy1)
```

```


Accuracy using naive bayes: 97.36842105263158


Accuracy using naive bayes: 91.2
```

In [18]:
```
cm=confusion_matrix(ytest,y_pred)
print("confusion matrix:\n",cm)
cm1=confusion_matrix(y1test,y_pred1)
print("confusion matrix:\n",cm1)
```

```
confusion matrix:
 [[12  0  0]
 [ 0 12  0]
 [ 0  1 13]]
confusion matrix:
 [[138   9]
 [ 13  90]]
```

In [27]:
```
fig,ax=plt.subplots(figsize=(4,4))
ax.imshow(cm)
ax.grid(False)
ax.xaxis.set(ticks=(0,1),ticklabels=("predicted yes","predicted no"))
ax.yaxis.set(ticks=(0,1),ticklabels=("Actual yes","Actual nos"))
ax.set_ylim(1.5,-0.5)
for i in range(2):
    for j in range(2):
        ax.text(j,i,cm[i,j],ha="center",va="center",color="red")
plt.show()
```



In [ ]:
```
```