import the necessary libraries

```
import numpy as np
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
```

create numpy array for actual and predicted labels

```
actual = np.array(
['Dog','Dog','Dog','Not Dog','Dog','Not Dog','Dog','Dog','Not Dog','Not Dog'])
predicted = np.array(
['Dog','Not Dog','Dog','Not Dog','Dog','Dog','Dog','Dog','Not Dog','Not Dog'])
```

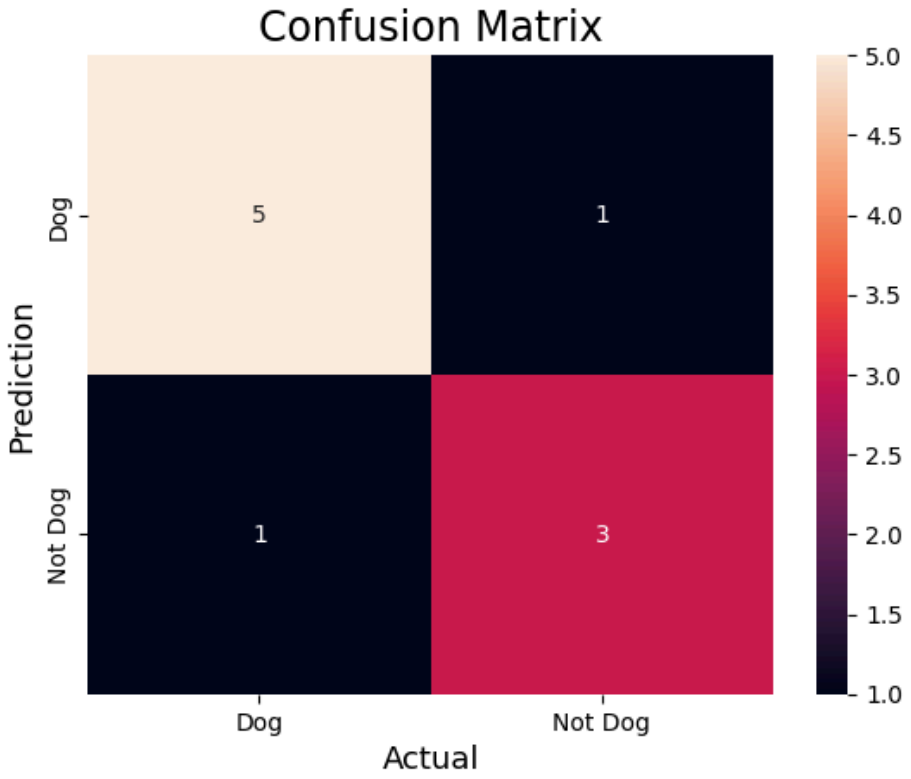
compute the confusion matrix

```
cm = confusion_matrix(actual,predicted)
```

Plot the confusion matrix with the help of the seaborn heatmap

```

sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels=['Dog','Not Dog'],
            yticklabels=['Dog','Not Dog'])
plt.ylabel('Prediction',fontsize=13)
plt.xlabel('Actual',fontsize=13)
plt.title('Confusion Matrix',fontsize=17)
plt.show()
```



Classifications Report based on Confusion Metrics

```
print(classification_report(actual, predicted))
```

```
              precision    recall  f1-score   support

         Dog       0.83      0.83      0.83         6
     Not Dog       0.75      0.75      0.75         4

    accuracy                           0.80        10
   macro avg       0.79      0.79      0.79        10
weighted avg       0.80      0.80      0.80        10
```

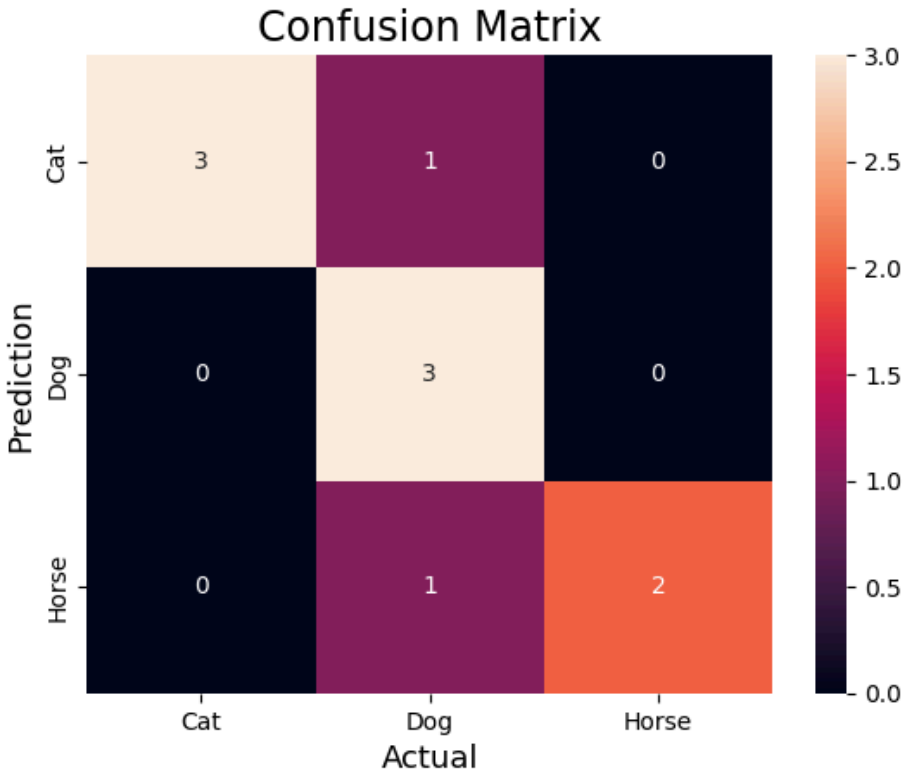
Implementation of Confusion Matrix for Binary classification using Python

```
import numpy as np
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
```

```
actual = np.array(
 ['Cat', 'Dog', 'Horse', 'Cat', 'Dog', 'Cat', 'Dog', 'Horse', 'Horse', 'Cat'])
predicted = np.array(
 ['Cat', 'Dog', 'Dog', 'Cat', 'Dog', 'Cat', 'Dog', 'Horse', 'Horse', 'Dog'])
```

```
cm = confusion_matrix(actual,predicted)
```

```
1 sns.heatmap(cm,
2    annot=True,
3    fmt='g',
4    xticklabels=['Cat', 'Dog', 'Horse'],
5    yticklabels=['Cat', 'Dog', 'Horse'])
6 plt.ylabel('Prediction', fontsize=13)
7 plt.xlabel('Actual', fontsize=13)
8 plt.title('Confusion Matrix', fontsize=17)
9 plt.show()
```




```
1   print(classification_report(actual, predicted))
```

	precision	recall	f1-score	support
Cat	1.00	0.75	0.86	4
Dog	0.60	1.00	0.75	3
Horse	1.00	0.67	0.80	3
accuracy			0.80	10
macro avg	0.87	0.81	0.80	10
weighted avg	0.88	0.80	0.81	10