PROG 3(a) CONFUSION MATRIX

AIM:

Evaluation matrix of ML algorithm using confusion matrix ; precision, accuracy, recall, specificity, sensitivity

SOURCE CODE:

```
import pandas as pd

import sklearn.linear_model as sk

import sklearn.model_selection as md

from sklearn import metrics


df = pd.read_csv("./iris.csv")

df = df.drop(df[df["variety"] == "Setosa"].index)

uni = df["variety"].unique()

df["variety"] = df["variety"].replace(uni, [0, 1])

x = df.iloc[:, :4]

y = df["variety"]


X_train, X_test, y_train, y_test = md.train_test_split(

   x, y, test_size=0.4,random_state=5)



logreg = sk.LogisticRegression()

logreg.fit(X_train, y_train)

y_pred = logreg.predict(x)


tm, fm = metrics.confusion_matrix(y, y_pred)

tn = tm[0]

fn = tm[1]

fp = fm[0]

tp = fm[1]
```

```
accuracy = (tp + tn)/(tp + tn + fp + fn)
sensitivity = tp/(tn + fn)
sensitivity = "{:2f}".format(sensitivity)
precision = tp/(tp + fp)
precision = "{:2f}".format(precision)
specificity = tn/(tn + fp)

print({"Accuracy": accuracy,
    "Precision": precision,
    "Sensitivity": sensitivity,
    "Specificity": specificity})
```

OUTPUT:

```
{'Accuracy': 0.97, 'Precision': '0.980000', 'Sensitivity': '0.980000', 'Specificity':
0.9795918367346939}
```

RESULT:

Thus the program was executed successfully and the output was verified.