Confusion matrix for a 3 class classification:

Let’s try to answer the above question with a popular dataset – IRIS DATASET.

The dataset has 3 flowers as outputs or classes, Versicolor, Virginia, Setosa.

With the help of petal length, petal width, sepal length, sepal width the model has to classify the given instance as Versicolor or Virginia or Setosa flower.

Let’s apply a classifier model here decision Tree classifier is applied on the above dataset. The dataset has 3 classes hence we get a 3 X 3 confusion matrix.

But how to know TP, TN, FP, FN values !!!!!

In the multi-class classification problem, we won’t get TP, TN, FP, FN values directly as in the binary classification problem.

We need to calculate for each class.

```
#importing packages
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']
dataset = pd.read_csv(url, names=names)
```

```
dataset
```

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
#To see first 5 rows of the dataset
dataset.head()
#To know the data types of the variables.
dataset.dtypes
```

```
sepal-length    float64
sepal-width     float64
petal-length    float64
petal-width     float64
Class            object
dtype: object
```

```
#Speceis is the output class,to know the count of each class we use value_counts()
dataset['Class'].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Class, dtype: int64
```

```
#Separating independant variable and dependent variable("Species")
X = dataset.drop(['Class'], axis=1)
y = dataset['Class']
# print(X.head())
print(X.shape)
# print(y.head())
print(y.shape)
```

```
(150, 4)
(150,)
```

```
# Splitting the dataset to Train and test
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

```
#to know the shape of the train and test dataset.
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(105, 4)
(105,)
(45, 4)
(45,)
```

```
#We use Support Vector classifier as a classifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
```

```
#training the classifier using X_Train and y_train
clf = SVC(kernel = 'linear').fit(X_train,y_train)
clf.predict(X_train)
```

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

```
#Testing the model using X_test and storing the output in y_pred
y_pred = clf.predict(X_test)
```

```
# Creating  a confusion matrix,which compares the y_test and y_pred
cm = confusion_matrix(y_test, y_pred)
```

```
# Creating a dataframe for a array-formatted Confusion matrix,so it will be easy for plotting
cm_df = pd.DataFrame(cm,
                     index = ['SETOSA','VERSICOLR','VIRGINICA'],
                     columns = ['SETOSA','VERSICOLR','VIRGINICA'])
```

```
cm_df
```

	SETOSA	VERSICOLR	VIRGINICA
SETOSA	16	0	0
VERSICOLR	0	17	1
VIRGINICA	0	0	11

```
#Plotting the confusion matrix
plt.figure(figsize=(5,4))
sns.heatmap(cm_df, annot=True)
plt.title('Confusion Matrix')
plt.ylabel('Actal Values')
plt.xlabel('Predicted Values')
plt.show()
```

1.Let us calculate the TP, TN, FP, FN values for the class Setosa using the Above tricks:

TP: The actual value and predicted value should be the same. So concerning Setosa class, the value of cell 1 is the TP value.

FN: The sum of values of corresponding rows except the TP value

FN = (cell 2 + cell3)

= (0 + 0)

= 0

FP : The sum of values of corresponding column except the TP value.

FP = (cell 4 + cell 7)

= (0 + 0)

= 0

TN: The sum of values of all columns and row except the values of that class that we are calculating the values for.

TN = (cell 5 + cell 6 + cell 8 + cell 9)

= 17 + 1 +0 + 11

= 29

Similarly, for Versicolor class the values/ metrics are calculated as below:

TP : 17 (cell 5)

FN : 0 + 1 = 1 (cell 4 +cell 6)

FP : 0 + 0 = 0 (cell 2 + cell 8)

TN : 16 +0 +0 + 11 =27 (cell 1 + cell 3 + cell 7 + cell 9).

I hope the concept is clear you can try for the Virginia class.