Given below is the 3 * 3 confusion matrix.

Predicted Value

	Class	**0**	**1**	**2**	**Total**
	0	34	13	5	**52**
Actual Value	**1**	0	52	0	**52**
	2	13	0	33	**46**
	Total	**47**	**65**	**38**	**150**

Find the Accuracy, Error, Recall, Precision, f1_score, f2_score, f_0.5_score, support, Micro f1, macro f1, weighted average and cohen's kappa.

Confusion Matrix:

A confusion matrix is a performance evaluation tool in machine learning, representing the accuracy of a classification model. It displays the number of true positives, true negatives, false positives, and false negatives. This matrix aids in analyzing model performance, identifying mis-classifications, and improving predictive accuracy.A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the total number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

Positive (P): Observation is positive.
Negative (N): Observation is negative.
True Positive (TP): Outcome where the model correctly predicts the positive class.
True Negative (TN): Outcome where the model correctly predicts the negative class.
False Positive (FP): Also called a type 1 error, an outcome where the model incorrectly predicts the positive class when it is actually negative.
False Negative (FN): Also called a type 2 error, an outcome where the model incorrectly predicts the negative class when it is actually positive.

Confusion Matrix for Multiple Class:

The concept of the multi-class confusion matrix is similar to the binary-class matrix. The columns represent the original or expected class distribution, and the rows represent the predicted or output distribution by the classifier.

Let us elaborate on the features of the multi-class confusion matrix with the above example.

******************************For Class 0***********************************

Unlike binary classification, there are no positive or negative classes in the multiclass. We have to find TP, TN, FP and FN for each individual class. For example, if we take class 0, then let's see what are the values of the metrics from the confusion matrix.

Let us consider, Class 0 is Positive value and Class 1 and 2 are Negative Value. Now with the help of binary classification.

Predicted

		Positive	**Negative**
Actual	**Positive**	TP	FN
	Negative	FP	TN

TP = 34

FP = 0 + 13 = 13

FN = 13 + 5 = 18

TN = 52 + 0 + 0 + 33 = 85

Accuracy:

This is simply equal to the proportion of predictions that the model classified correctly.

Accuracy of class 0 = (TP + TN) / (TP + FP + FN + TN)

= (34 + 85) / (34 + 13 + 18 + 85)

= 0.79

Error:

Error of class 0 = 1 - Accuracy

= 1 - 0.79

= 0.21

Precision:

Precision is also known as positive predictive value (PPV) and is the proportion of relevant instances among the retrieved instances. In other words, it answers the question “What proportion of positive identifications was actually correct?”

Precision of class 0 = TP / (TP + FP)
= 34 / (34 + 13)
= 0.72

Recall:

Recall, also known as the sensitivity, hit rate, or the true positive rate (TPR), is the proportion of the total amount of relevant instances that were actually retrieved. It answers the question “What proportion of actual positives was identified correctly?”

Recall of class 0 = TP / (TP + FN)
= 34 / (34 + 18)
= 0.65

F1_score:

The F1 score is a measure of a test’s accuracy—it is the harmonic mean of precision and recall. It can have a maximum score of 1 (perfect precision and recall) and a minimum of 0. Overall, it is a measure of the preciseness and robustness of your model.

F_1 of class 0 = 2 * (precision_0 * recall_0) / (precision_0 + recall_0)
= 2 * 0.72 * 0.65 / (0.72 + 0.65)
= 0.68

F0.5_score:

F_0.5 of class 0 = 1.25 * (precision_0 * recall_0) / (0.25 * precision_0 + recall_0)
= 1.25 * 0.72 * 0.65 / (0.25 * 0.72 + 0.65)
= 0.70

F2_score:

F_2 of class 0 = 5 * (precision_0 * recall_0) / (4 * precision_0 + recall_0)
= 5 * 0.72 * 0.65 / (4 * 0.72 + 0.65)
= 0.663

Support:
Suppor the sum of actual values of each classes.

Support of class 0 = 34 + 13 + 5 = 52

**************************Similarly for class 1*************************************

TP = 52

FP = 13 + 0 = 13

FN = 0 + 0 = 0

TN = 34 + 5 + 13 + 33 = 85

Accuracy:

Accuracy of class 1 = (TP + TN) / (TP + FP + FN + TN)

= (52 + 85) / (52 + 13+ 0+ 85)

= 0.91

Error:

Error of class 1 = 1 - Accuracy

= 1 - 0.91

= 0.09

Precision:

Precision of class 1 = TP / (TP + FP)

= 52/ (52 + 13)

= 0.8

Recall:

Recall of class 1 = TP / (TP + FN)
= 52/ (52 + 0)
= 1

F1_score:

F_1 of class 1 = 2 * (precision_1 * recall_1) / (precision_1 + recall_1)
= 2 * 0.8 * 1 / (0.8 + 1)
= 0.89

F0.5_score:

F_0.5 of class 1 = 1.25 * (precision_1 * recall_1) / (0.25 * precision_1 + recall_1)

= 1.25 * 0.8 * 1 / (0.25 * 0.8 + 1)
= 0.83

F2_score:

F_2 of class 1 = 5 * (precision_1 * recall_1) / (4 * precision_1 + recall_1)
= (5 * 0.8 * 1) / (4 * 0.8 + 1)
= 0.95

Support:

Support of class 1 = 0 + 52 + 0 = 52

********************************Similarly for class 2********************************

TP = 33

FP = 0 + 5 = 5

FN = 0 + 13 = 13

TN = 34 + 13 + 0 + 52 = 99

Accuracy:

Accuracy of class 2 = (TP + TN) / (TP + FP + FN + TN)
= (33 + 99) / (33 + 5+ 13+ 99)
= 0.88

Error:

Error of class 2 = 1 - Accuracy
= 1 - 0.88
= 0.12

Precision:

Precision of class 2 = TP / (TP + FP)
= 33/ (33 + 5)
= 0.868

Recall:

Recall of class 2 = TP / (TP + FN)

= 33/ (33 + 13)
= 0.717

F1_score:

F_1 of class 2 = 2 * (precision_2 * recall_2) / (precision_2 + recall_2)
= 2 * 0.868 * 0.717 / (0.868 + 0.717)
= 0.785

F0.5_score:

F_0.5 of class 2 = 1.25 * (precision_2 * recall_2) / (0.25 * precision_2 + recall_2)
= 1.25 * 0.868 * 0.717 / (0.25 * 0.868 + 0.717)
= 0.833

F2_score:

F_2 of class 2 = 5 * (precision_2 * recall_2) / (4 * precision_2 + recall_2)
= 5 * 0.868 * 0.717 / (4 * 0.868 + 0.717)
= 0.743

Support:

Support of class 2 = 13 + 0 + 33 = 46

From above calculations,

Class	Accuracy	Error	Precision	Recall	F_1 score	F_0.5 score	F_2 score	support
0	0.79	0.21	0.72	0.65	0.68	0.7	0.663	52
1	0.91	0.09	0.8	1	0.89	0.83	0.95	52
2	0.88	0.12	0.868	0.717	0.785	0.833	0.743	46

Macro f_1:
It takes the f1 score of each class and averages them, treating each classes equally.

Micro f_1 = (f_1 of class 0 + f_1 of class 1 + f_1 of class 2) / 3
= (0.68 + 0.89 + 0.785) / 3
= 0.785

Weighted-average:
Weighted-average computes the metric for each class independently and then takes the average across all classes, weighted by the support (number of true instances) for each class.It gives higher weight to classes with more instances, meaning classes with higher support have a greater impact on the overall average.To compute weighted-average precision, recall, or F1 score, you calculate the metric for each class and then compute a weighted average using the support for each class as weights.

weighted *average* = class_0_f1 * (support_0 / total) + class_1_f1 * (support_1 / total) + class_1_f1 * (support_1 / total)
= 0.68 * (52 / 150) + 0.89 * (52 / 150) + 0.785 * (46 / 150)
= 0.785

Micro_f1:

overall _TP(TP) = 119
overall_FP(FP) = 31
overall_FN(FN) = 31

Micro_f1 = TP / TP + 1/2(FP + FN)
= 119 / (119 + 1/2*(31 + 31))
= 0.793

Cohen's Kappa:
The Cohen-Kappa score can be used to measure the degree to which two or more raters can diagnose, evaluate, and rate behavior