CLASS WISE PERFORMANCE EVALUATION MEASURES BY CALCULATING VARIOUS FACTORS USING CONCEPT OF CONFUSION MATRIX

Shanmukha Sudha Kiran.T-**211FA04003**, Sairam.P - **211FA04284**,

Sripujitha .S -**211FA04289**, Vamsi.G - **211FA04358**

BATCH-07,3Rd B. TECH, CSE BRANCH, VFSTR DEEMED TO BE UNIVERSITY

ABSTRACT:

In this Paper we begin by introducing about confusion matrix, confusion matrix has been provided to evaluate the performance of a classifier. The matrix depicts the predicted classes against the true classes. From this, various performance evaluation measures have been calculated.

PROBLEM STATEMENT:

A dataset has three class labels, namely Apple, Orange and Mango. The following is a possible confusion matrix for these classes.

		True class		
		APPLE	ORANGE	MANGO
Predicted Class	APPLE	8	6	7
	ORANGE	2	6	2
	MANGO	4	2	3

For reference: Accuracy = (true positives + true negatives) / (total examples) Precision = (true positives) / (true positives + false positives) Recall = (true positives) / (true positives + false negatives) F1 score = (2 * precision * recall) / (precision + recall) Find the class wise performance evaluation measures and analyze which class accuracy is more out three classes. Also, measure the algorithm for classifier's F1 score (as a value form 0 to 1.

INTRODUCTION:

CONFUSION MATRIX:

A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data.

ACCURACY:

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition: Accuracy = Number of correct predictions Total number of predictions.

Accuracy = (true positives + true negatives) / (total examples)

PRECISION:

Precision is one indicator of a machine learning model's performance – the quality of a positive prediction made by the model.

Precision = (true positives) / (true positives + false positives)

RECALL:

Recall, also known as the true positive rate (TPR), is the percentage of data samples

that a machine learning model correctly identifies as belonging to a class of interest—the “positive class”—out of the total samples for that class.

Recall = (true positives) / (true positives + false negatives)

F1-SCORE:

F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model.

F1 score = (2 * precision * recall) / (precision + recall)

ALGORITHM:

1.start

2. Import necessary libraries if needed.

3. Initialize variables:

- Create an empty dictionary `d` to store class indices and names.

- Create an empty dictionary `e` to store accuracy values and corresponding class indices.

- Initialize `tp`, `tn`, `fp`, and `fn` to 0 for counting.

4. Input the number of classes (`m`) from the user.

5. Input class names and store them in the dictionary `d`:

- Loop over `m` times:

- Input a class name and store it in the dictionary with its index.

6. Initialize an empty list `matrix` to store the confusion matrix values.

7. Input confusion matrix values for each class:

- Loop over `m` times for each row:

- Initialize an empty list `l` for the current row.

- Loop over `m` times for each column:

- Input a confusion matrix value and append it to list `l`.

- Append the list `l` to the matrix.

8. For each class index `i` (0 to `m-1`):

- Calculate true positives (`tp`):

- Assign the value from `matrix[i][i]` to `tp`.

- Initialize `fp` and `fn` to 0 for counting false positives and false negatives.

9. Nested loops for calculating false positives (`fp`), false negatives (`fn`), and true negatives (`tn`):

- Loop over `m` times for each predicted class `j`:

- If `i` is not equal to `j`:

- Increment `fp` by `matrix[i][j]`.

- Increment `fn` by `matrix[j][i]`.

- Loop over `m` times for each actual class `k`:

- Increment `tn` by `matrix[j][k]`.

- Adjust `tn` by subtracting the sum of `tp`, `fp`, and `fn`.

10. Calculate accuracy using the `accuracy` function:

- Calculate `acc` using the provided formula and values (`tp`, `tn`, `fp`, `fn`).

11. Calculate precision using the `precision` function:

- Calculate `pre` using the provided formula and values (`tp`, `fp`).

12. Calculate recall using the `recall` function:

- Calculate `rec` using the provided formula and values (`tp`, `fp`).

13. Calculate F1 score using the `f1score` function:

- Calculate `score` using the provided formula and values (`pre`, `rec`).

14. Print evaluation measures for the current class:

- Print TP, TN, FP, FN, accuracy, precision, recall, and F1 score.

- Reset counters `tp`, `tn`, `fp`, and `fn` to 0.

15. After looping through all classes, find the class with the maximum accuracy:

- Determine the class index with the maximum accuracy from dictionary `e`.

- Print the class name with the maximum accuracy.

16.Stop

SOURCE CODE:

```
def accuracy(tp,tn,fp,fn):
    acc=(tp+tn)/(tp+tn+fp+fn)
    return acc
def precision(tp,fp):
    pre=tp/(tp+fp)
    return pre
def recall(tp,fp):
    rec=tp/(tp+fn)
    return rec
def f1score(pre,rec):
    score=(2*pre*rec)/(pre+rec)
    return score


m=int(input("enter the no of classes: "))
matrix=[]
d={}
e={}
tp=tn=fp=fn=0
print("Enter the names of the class: ")
for i in range(m):
    p=input()
    d[i]=p
print("Now enter the values:")
for i in range(m):
    l=[]
    for j in range(m):
        n=int(input())
        l.append(n)
    matrix.append(l)
for i in range(m):
    tp=matrix[i][i]
    for j in range(m):
        if(i!=j):
            fp=fp+matrix[i][j]
            fn=fn+matrix[j][i]
        for k in range(m):
            tn=tn+matrix[j][k]
    tn=tn-(tp+fp+fn)
    acc=accuracy(tp,tn,fp,fn)
    e[acc]=i
```

```
    pre=precision(tp,fp)
    rec=recall(tp,fp)
    score=f1score(pre,rec)
    print("********************for class ",d[i]," values are: ")
    print()
    print("tp tn fp tn :",tp ,tn ,fp ,fn)
    print("Accuracy is: ",acc)
    print("precision is: ",pre)
    print("Recall is: ",rec)
    print("F1 Score is: ",score)
    print()
    tp=tn=fp=fn=0
print("Class with maximum accuracy is :")
print(d[e[max(e)]])
```

OUTPUTS:

```
Enter the names of the class:
apple
orange
mango
Now enter the values:
8
6
7
2
6
2
4
2
3
********************for class  apple  values are:

tp tn fp tn : 8 13 13 6
Accuracy is:  0.525
precision is:  0.38095238095238093
Recall is:  0.5714285714285714
F1 Score is:  0.4571428571428571

********************for class  orange  values are:

tp tn fp tn : 6 22 4 8
Accuracy is:  0.7
precision is:  0.6
Recall is:  0.42857142857142855
F1 Score is:  0.5

********************for class  mango  values are:

tp tn fp tn : 3 22 6 9
Accuracy is:  0.625
precision is:  0.3333333333333333
Recall is:  0.25
F1 Score is:  0.28571428571428575

Class with maximum accuracy is :
orange
```

CONCLUSION:

In conclusion, the evaluation measures reveal that the Orange class exhibits the highest accuracy, precision, and recall among the three classes. Furthermore, the F1 score reinforces the Orange class's strong performance as a classifier.

REFERENCE:

1) https://www.geeksforgeeks.org/confusion-matrix-machine-learning/
2) https://www.javatpoint.com/confusion-matrix-in-machine-learning