Confusion Matrix for Multi-Class Classification

BEGINNER CLASSIFICATION MACHINE LEARNING PYTHON

This article was published as a part of the Data Science Blogathon

Introduction

Confusion Matrix is used to know the performance of a Machine learning classification. It is represented in a matrix form.

Confusion Matrix gives a comparison between Actual and predicted values.

The confusion matrix is a N x N matrix, where N is the number of classes or outputs.

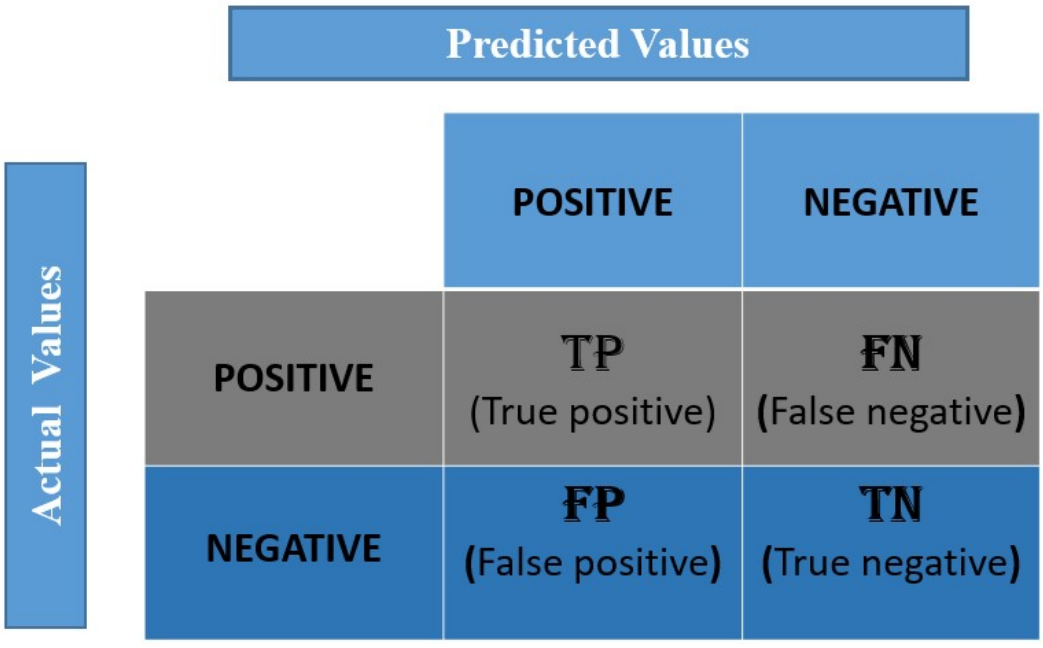
For 2 class ,we get 2 x 2 confusion matrix.

For 3 class ,we get 3 X 3 confusion matrix.

Confusion Matrix has 4 terms to understand True Positive(TP),False Positive(FP),True Negative(TN) and False Negative(FN).

How does the confusion matrix look like?

Below is the representation of the confusion matrix.



Confusion matrix for Binary Classification

Let us understand the confusion matrix for a simple binary classification example.

Binary classification has 2 outputs, the inputs for this classification will fall in either of the 2 outputs or classes.

Example: Based on certain inputs we have to decide whether the person is sick or not, diabetic or not.

Let us see how to construct a confusion matrix and understand its terminologies. Consider we have to model a classifier that classifies 2 kinds of fruits.

We have 2 types of fruits Apples and grapes and we want our machine learning model to identify or classify the given fruit as an Apple or grapes.

So we take 15 samples of 2 fruits, out of which 8 belong to Apples and 7 belong to the grapes class. (Class is nothing but the output, in this example, we have 2 output classes apple and grapes). We will represent Apple as 1 and grape as 0 class.

The actual class for 8 apples and 7 grapes,can be represented as

Actual = [1,1,1,1,1,1,1,1,0,0,0,0,0,0,0]

The classifier model to predict 1 for Apple and 0 for grape.

Assume that the classifier takes all the 15 inputs and, makes the following predictions:

· Out of 8 apples, it will classify 5 correctly as Apple and wrongly predict 3 as grapes.

· Out of 7 grapes, it will classify 5 correctly as grapes and wrongly predicts 2 as an apple.

The prediction of the classifier may be as:

Prediction = [1,0,0,0,1,1,1,1,0,0,0,0,0,1,1]

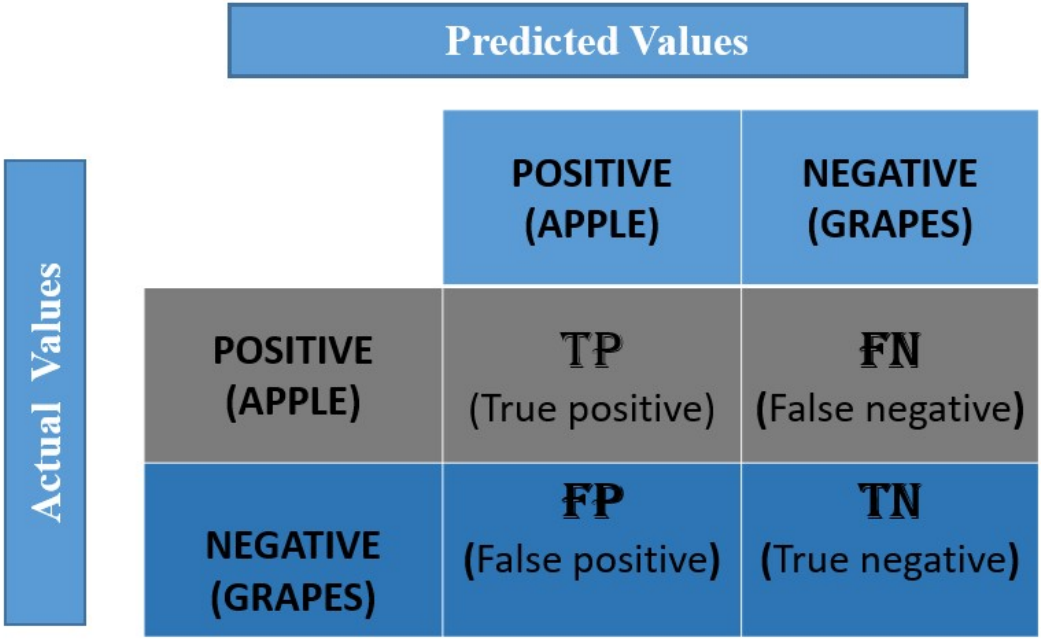
```
# Creation of confusion matrix in using sklearn from sklearn.metrics import confusion_matrix


#Let actual value be 1 for apple and 0 grapes for our example ACTUAL = [1,1,1,1,1,1,1,1,0,0,0,0,0,0,0] # Let the predicted values be PREDICTION= [1,0,0,0,1,1,1,1,0,0,0,0,0,1,1]


# Confusion matrix for actual and predicted values. matrix = confusion_matrix(ACTUAL,PREDICTION, labels= [1,0]) print('Confusion matrix : n',matrix)


# outcome values order in sklearn TP, FN, FP, TN = confusion_matrix(ACTUAL,PREDICTION,labels= [1,0]).reshape(-1) print('Outcome values : n', TP, FN, FP, TN)
```

The confusion matrix for this example can be visualized as below.



For our example the positive value is Apple and the negative value is Grapes.

True Positive:

It means the actual value and also the predicted values are the same.

In our case the actual value is also apple and the model prediction is also apple.

If you observe for the TP cell the positive value is the same for Actual and predicted.

False Negative:

This means the actual value is positive in our case it is apple but the model has predicted it as negative i.e., grapes. So the model has given the wrong prediction, It was supposed to give a positive(apple) but it has given a negative(grape) so whatever the negative output we got is false, hence the name False Negative.

False Positive:

This means the actual value is negative in our case it is grapes but the model has predicted it as positive i.e., apple. So the model has given the wrong prediction, it was supposed to give negative(grape) but it has given positive(apple) so whatever the positive output we got is false, hence the name False Positive.

True Negative:

It means the actual value and also the predicted values are the same. In our case, the actual values are also grapes and the Prediction is also Grapes.

The values for the above example are:

TP=5, FN=3, FP=2, TN=5.

Confusion matrix for Multi-class classification.

The above example is a binary classification with only 2 outputs so we got a 2 X 2 matrix.

So what if the outputs are greater than 2 classes i.e., Multi-class classification.

How to calculate TP, FN, FP, TN?

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.

Source: Google

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


#Importing of dataset to dataframe. df = pd.read_csv("../input/iris-flower-dataset/IRIS.csv")


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


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


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

```
# 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)


#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)


#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'])


#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()
```

How to calculate FN, FP, TN, TP :

FN: The False-negative value for a class will be the sum of values of corresponding rows except for the TP value.FP: The False-positive value for a class will be the sum of values of the corresponding column except for the TP value.TN: The True Negative value for a class will be the sum of values of all columns and rows except the values of that class that we are calculating the values for.

TP: The True positive value is where the actual value and predicted value are the same.

The confusion matrix for the IRIS dataset is as below:

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.

Why Confusion matrix?

Confusion Matrix allows us to measure Recall, Precision,
Accuracy and AUC-ROC curve are the metrics to measure the performance of the model.

Thanks for reading :)

If there are any doubts and suggestions feel free to connect with me through email:cbharathids@gmail.com

Note: The images are created by the Author except for the one which has the source mentioned.

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

Article Url - https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/

cb7705