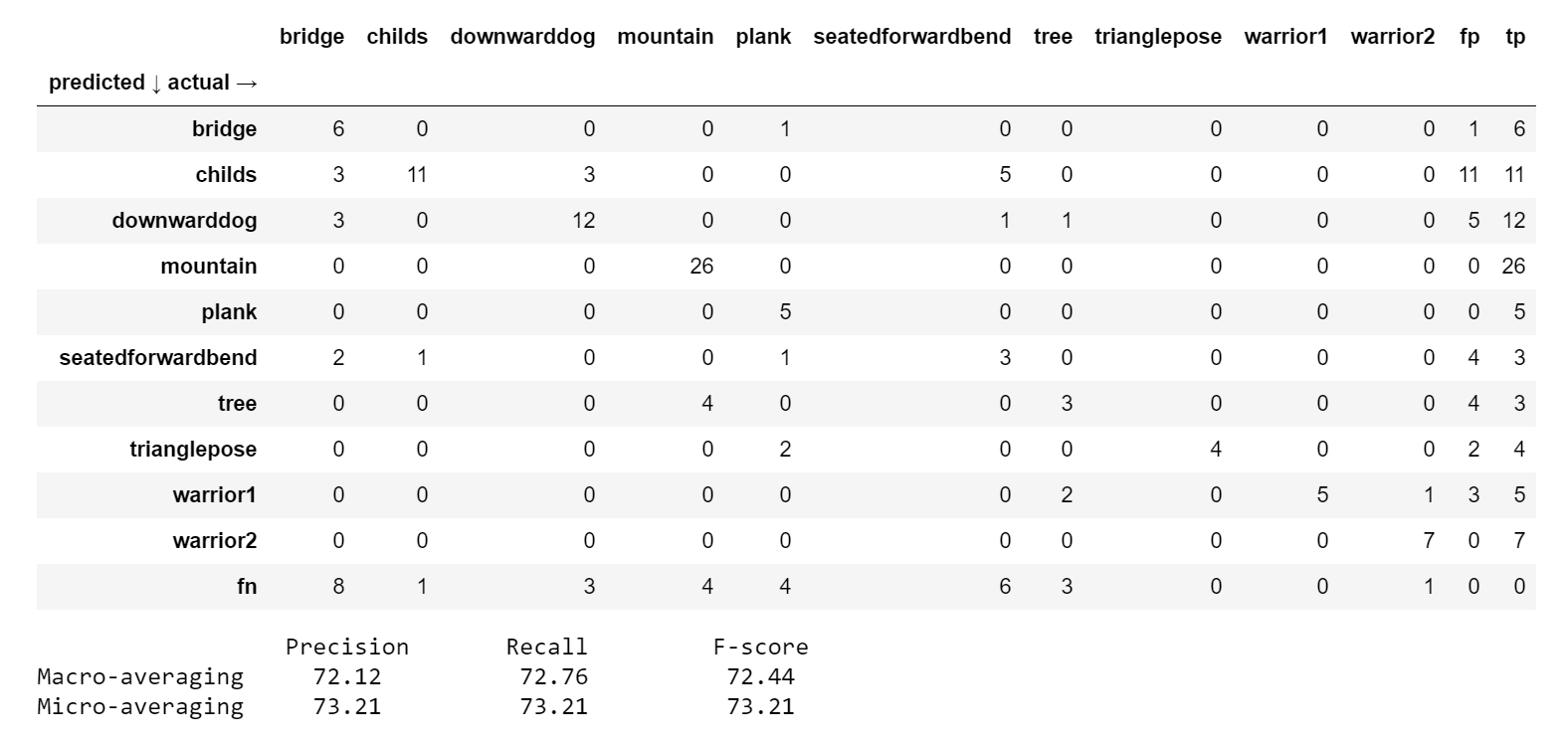
**ASSIGNMENT 1**

**STUDENT ID: 1013239 & 1012861**

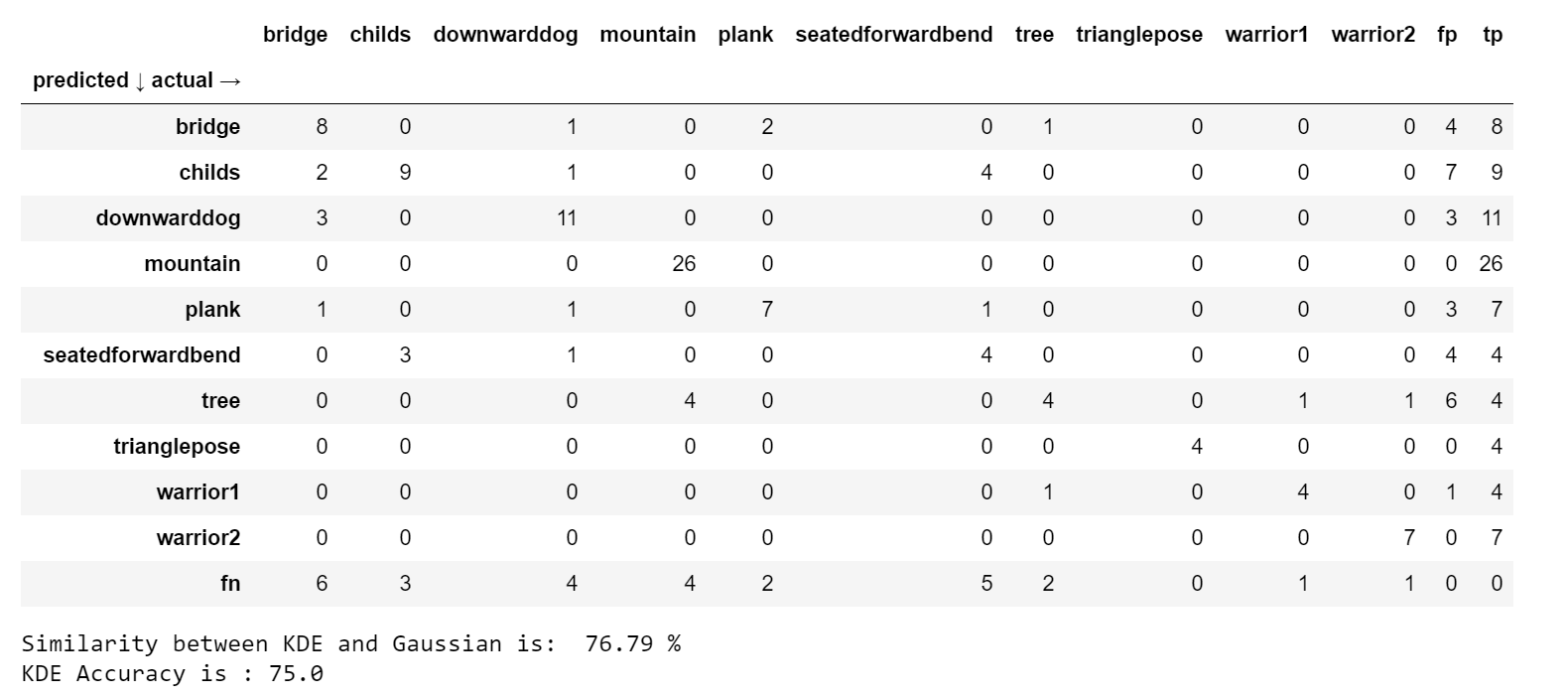
***QUESTION 1***

By trying both macro and micro averaging, we found that there is a higher precision, recall, and f-score value when using the micro averaging method. This is because macro averaging takes the average precision and recall values of each class while micro averaging sums the true positives of each instance in a class and divides that by the sum of the true positives and false positives, to get precision, or false negatives, to get recall. This causes the precision and recall value of micro-averaging to also be equal since theoretically every false positive of a class will be a false negative of another class. Macro-averaging shows a lower result than micro-averaging due to class imbalances which means that each value is treated equally where there are cases where a class only contains a smaller number of instances compared to another (e.g. ‘mountain’ class has 160 instances where ‘seatedforwardbend’ class only has 42 instances). This effect can be minimized by using the weighted averaging method in which each class is given different weights according to the number of instances represent a particular class. This should result in a precision and recall number that is closer to the value obtained through micro-averaging.



***QUESTION 3***

As an arbitrary kernel bandwidth, we chose to use 10 since there are 10 different classes in the dataset. The accuracy obtained through the Gaussian implementation was 73% while in the KDE implementation we got an accuracy increase with 75%. This might suggest that some of the numerical data does not follow a Gaussian distribution, and hence the assumption that it does is false leading to lower accuracy values. The predicted classes obtained from the KDE implementation is 76.8% similar to the one obtained by the Gaussian distribution. From calculating the values of false positives and negatives obtained from the KDE implementation, we can see that there is a decrease in the number of false positives for some attributes such as ‘childs’ however an increase for other attributes such as ‘bridge’, suggesting that attributes such as ‘childs’ do not follow a Gaussian distribution while attributes like ‘bridge’ can be said to follow a Gaussian distribution. However, since the increase in accuracy is not that significant (approximately 3%), we can deduce that this data is approximately Gaussian. This is a good assumption to make since it can reduce computation time resulting in increased computational efficiency especially for larger test sets.



***QUESTION 4***

In this question, we use cross-validation to choose the kernel bandwidth. We split the training data into 8 partitions to depict the ratio of the actual training dataset : test set of 747 : 116 ≈ 7 : 1. Within each partition, we try different kernel bandwidths of 5,10,15,20,25 from the recommended 5-25 range, where we will choose the kernel which results in the highest accuracy. Once KDE Naïve Bayes is done using different kernel bandwidths within each partition, we take the average of the best bandwidths chosen. We will then perform KDE Naïve Bayes on the actual test dataset using the averaged kernel bandwidths.

Our cross validation for choosing kernel bandwidths chooses the bandwidth which results in the highest accuracy and our implementation found that 5.0 is the best choice with an accuracy of 77.68% compared to using the chosen arbitrary kernel bandwidth of 10 in which it results in an accuracy of 75%. As can be seen from the graph below for class bridge with x1 vs its KDE, when bandwidth = 5 (red), more peaks can be seen when compared to bandwidth = 5 (blue) which results in better generalisation of the classes.

Chart, histogram

Description automatically generated

Some peaks seen when kde = 5 (red) and not seen in kde = 10(blue)

***QUESTION 5***

Here, we use mean imputation to replace the missing values in certain attributes. However, we found that despite this makes the mean of the class higher and less skewed due to the 0-imputation method we initially used, the accuracy is exactly the same as if the missing values is ignored when using the Gaussian implementation. When we inputted into the KDE classifier, it shows a significant decrease in accuracy, from 75% to 69%. This might suggest that the mean-imputed data is closer to resembling a Gaussian distribution instead of an unknown distribution hence allowing the accuracy of the Gaussian Naïve Bayes classifier to stay the same but reducing the accuracy for the KDE classifier. This data suggest that missing values are useful in this task and that it should be left empty instead of trying to impute it with mean values. This is because here the missing values are intentional, to indicate that the hand is placed there to complete the pose and hence the data should not be altered.

