Lab 1 – Introduction to Machine Learning

Author: Gonçalo Aguiar 904475

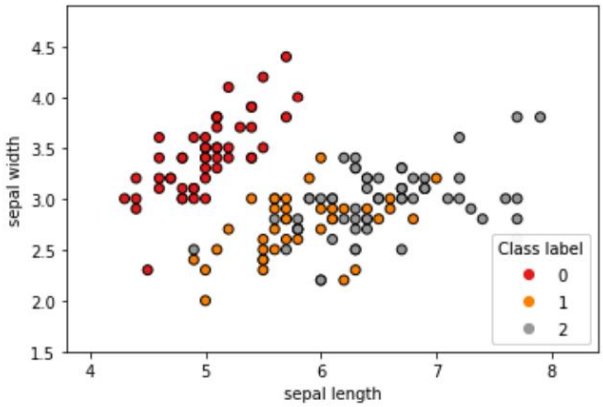
1. **Check the dimensions of the X and Y arrays**

To check the dimensions of the X and Y arrays the following code was used:

```
iris = datasets.load_iris()
X = iris.data
Y = iris.target
print(len(X))
print(len(Y))
```

The printed result was 150 for each of them as expected.

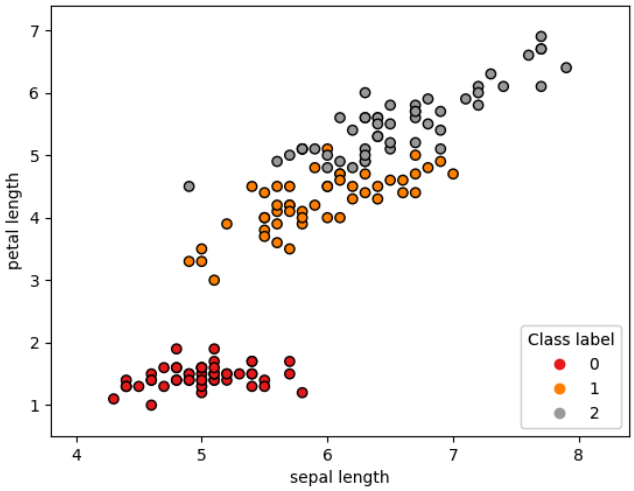
2. **Analyze the chart. Do the two visualized features allow for unambiguous classification of the Iris species based on them? Modify the chart so that it includes two different features.**



No, just with the two first visualized features(sepal width and sepal length) there is ambiguity because although we can separate class 0 from class 1 and class 0 from class 2(we can draw a line like y =mx + b), we can't to the same thing with class 1 and class 2 because the results overlap as we can see in the plot.

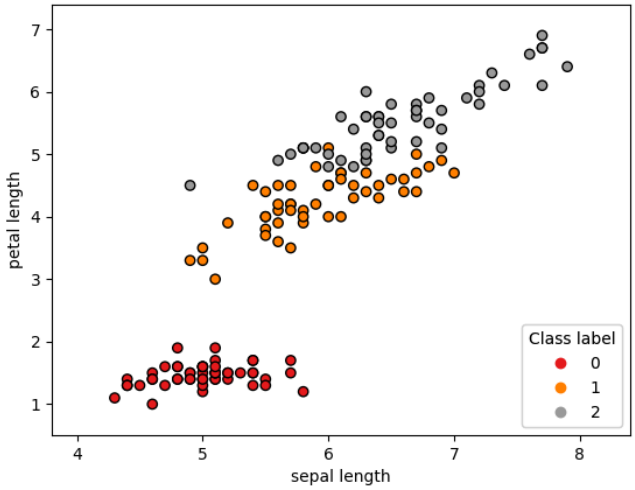
In order to include two diferent features the following code was changed and the result was the next plot.

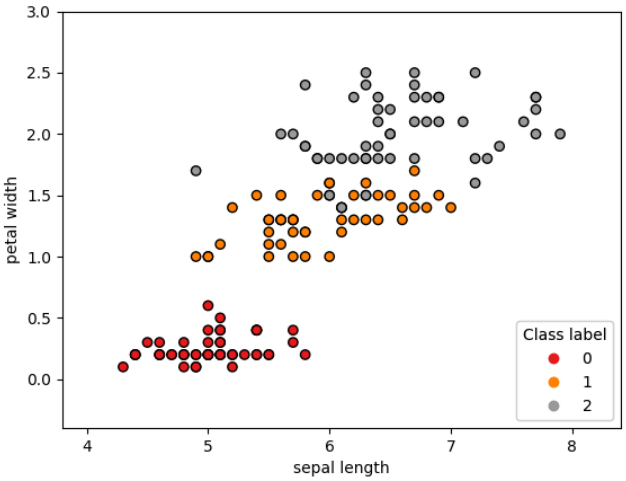
```
x_min, x_max = X[:,0].min() - 0.5 , X[:,0].max() + 0.5
y_min, y_max = X[:,2].min() - 0.5 , X[:,2].max() + 0.5


fig, ax = plt.subplots()

scatter = ax.scatter(X[:, 0], X[:, 2],c=Y,cmap=plt.cm.Set1,edgecolor= "k")
```

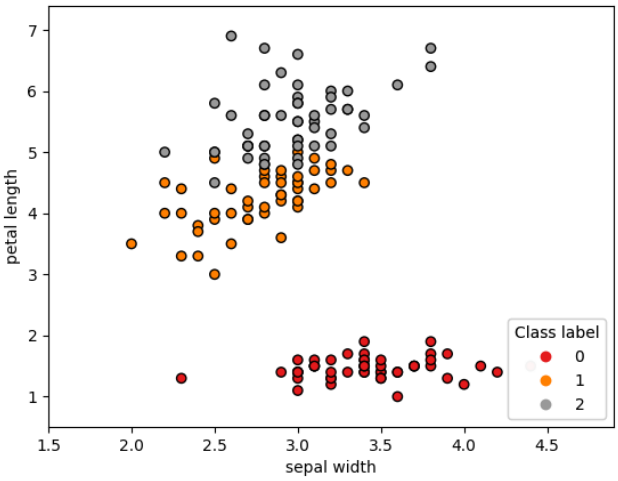


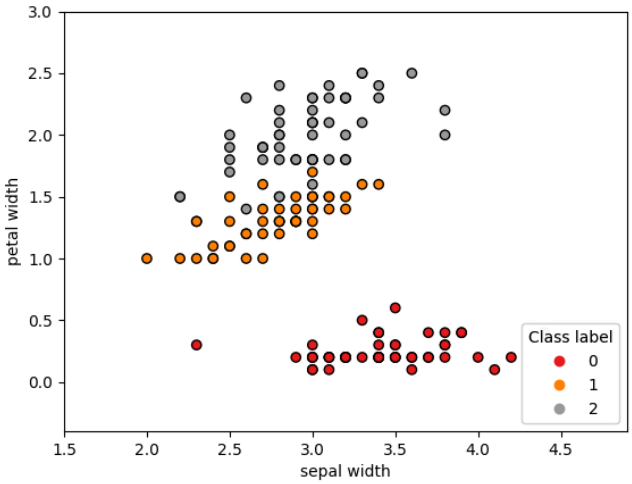
3. **Modify the code to display other combinations of features (with 4 features, there are 6 different pairs).**

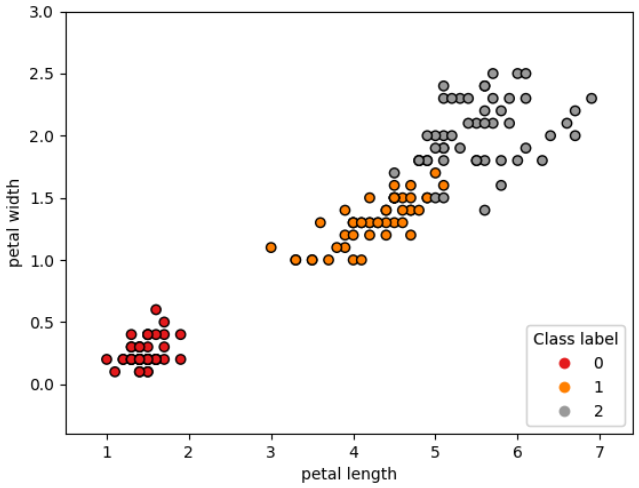
   The resulting plots for the 6 combinations are the following. As we can see in the plots there is always some overlap between classes 1 and 2. Class 0 can be always separated from the two others.

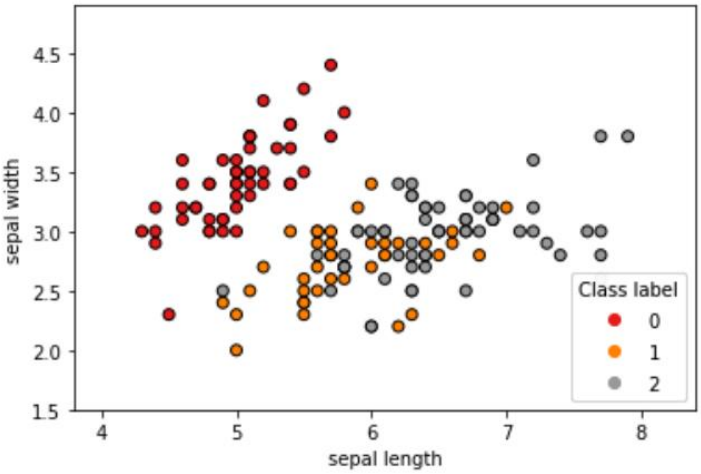









petal width
petal length
Class label
0
1
2

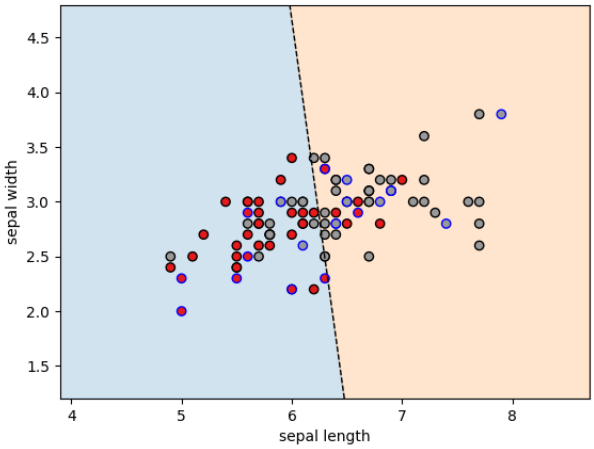

sepal width
sepal length
Class label
0
1
2

4. **Modify the code accordingly to train a classifier that distinguishes these two classes based on the same features - the length and width of the sepals. Save the resulting plot with samples and decision boundary. Count, for both classes, how many training and test samples were misclassified. After completing this task, run the code again, which will cause a new random split of samples into training and test sets. See if the decision boundary has changed. Do classification results depend on the choice of the training set?**

   In order to change from comparing class 0 and class 1 to comparing class 1 to class 2 the following code was changed.

```
X01_train = np.concatenate([X1[:,0:2], X2[:,0:2]])
Y01_train = np.concatenate([Y1,Y2])
X01_test = np.concatenate([X1_test[:,0:2], X2_test[:,0:2]])
Y01_test = np.concatenate([y1_test,y2_test])
```

   The first resulting plot was the following. As we can see in the plot there are in total 4 misclassified test samples (2 in class 1 and 2 in class 2) and there are in total 13 misclassified training samples (5 in class 1 and 8 in class 2).



   After running the code again the resulting plot was the following. If we can compare the decision boundaries we can see that they are different. Since the training set is different in both cases we can reach the conclusion that the classification results depend of the training set.