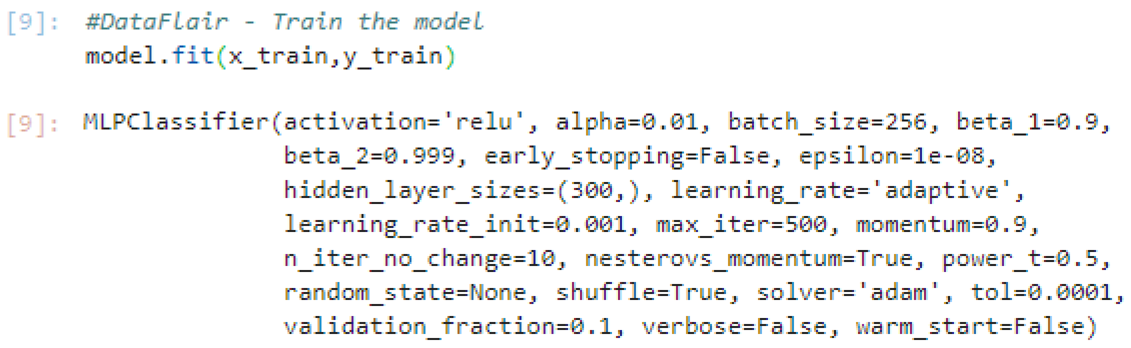
Output Screenshot:



```
[9]: #DataFlair - Train the model
     model.fit(x_train,y_train)

[9]: MLPClassifier(activation='relu', alpha=0.01, batch_size=256, beta_1=0.9,
                   beta_2=0.999, early_stopping=False, epsilon=1e-08,
                   hidden_layer_sizes=(300,), learning_rate='adaptive',
                   learning_rate_init=0.001, max_iter=500, momentum=0.9,
                   n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
                   random_state=None, shuffle=True, solver='adam', tol=0.0001,
                   validation_fraction=0.1, verbose=False, warm_start=False)
```

10. Let's predict the values for the test set. This gives us y_pred (the predicted emotions for the features in the test set).

```
#DataFlair - Predict for the test set
y_pred=model.predict(x_test)
```

Screenshot:

11. To calculate the accuracy of our model, we'll call up the accuracy_score() function we imported from sklearn. Finally, we'll round the accuracy to 2 decimal places and print it out.

```
#DataFlair - Calculate the accuracy of our model
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)

#DataFlair - Print the accuracy
print("Accuracy: {:.2f}%".format(accuracy*100))
```

Output Screenshot:

Summary

In this Python mini project, we learned to recognize emotions from speech. We used an MLPClassifier for this and made use of the soundfile library to read the sound file, and the librosa library to extract features from it. As you'll see, the model delivered an accuracy of 72.4%. That's good enough for us yet.

Hope you enjoyed the mini python project.