

Report:

The purpose of this code is to teach the computer to understand and classify text into two categories. One is S and another is L. The program takes a bunch of text and looks at different features like the number of sentences, words, and how often certain words appear and use it to train 2 models. Logistic and Multi-Layer Perceptron.

After running the program successfully, I got the following result:

Logistic Regression Accuracy: 1.00

MLP Accuracy: 0.44

Logistic Regression Precision: 1.00

MLP Precision: 0.75

Logistic Regression Recall: 1.00

MLP Recall: 0.44

Logistic Regression F1 Score: 1.00

MLP F1 Score: 0.37

The Logistic Regression model performs exceptionally well with perfect accuracy, precision, recall, and F1 score. It can accurately classify all test samples.

On the other hand, the MLP model struggles with an accuracy of 44% and relatively lower precision, recall, and F1 score. It seems to have difficulty generalizing to the test data.

The high performance of the Logistic Regression model suggests that the features, particularly the TF-IDF weighted word frequencies, used in the model are perfect to separate the two classes in the training data.

But there's a little worry: because Logistic Regression is doing so well, we should check if it's maybe too focused on the training data and not good at understanding new, different texts to verify that it is not "overfitting,"

Can Any Feature Give 100% Accuracy:

Usually, it's hard to find a single thing that can make a model 100% accurate.

Text is tricky because it can be written in many different ways, and sometimes there are no clear patterns that can guarantee 100% accuracy.