

The first model has an accuracy of 0.5, which is equivalent to random guessing, indicating that the model is not performing well. However, the high recall score also indicates that the model is producing a high number of false positives, which can be addressed by adjusting the model's threshold or using other evaluation metrics such as the area under the ROC curve (AUC-ROC). Both word2vec and AraVec vectors have their own advantages and disadvantages, and the performance of the ANN model depends on the quality of the pre-trained vectors and the complexity of the text data. It's worth noting that the precision score for the second model is affected by the UndefinedMetricWarning message, which indicates that precision is ill-defined and being set to 0.0 in labels with no predicted samples. Both models have low accuracy, it could be due to the fact that the text data is very complex or noisy, and the pre-trained vectors are not able to capture the semantic meaning of the words. The precision is also higher at 0.55, indicating that the model is producing fewer false positives. In summary, the second model using AraVec representation has a better performance compared to the first model, as indicated by the higher accuracy, precision, recall, and F1 score. If the ANN model with AraVec representation has a higher training time than the ANN model with word2vec representation, it could be due to the fact that AraVec vectors are not as optimized as word2vec vectors. However, the low precision indicates that the model is also producing a high number of false positives. Therefore, loading and processing AraVec vectors might take more time.