**Performance:**

**Evaluation:**

We evaluated the performance of each model using the following metrics:

Accuracy: The proportion of correctly classified articles

Precision: the proportion of true positives out of all predicted positives

Recall: the proportion of true positives out of all actual positives

F1-score: the harmonic mean of precision and recall

ROC Curve and AUC: the plot of the true positive rate against the false positive rate, and the area under the curve (AUC). We plotted the ROC curve and calculated the AUC to measure the model's ability to discriminate between real and fake news.

**Results:**

After training and evaluating our models, we found that all our models produced a great accuracy score, grid best score and great validation results, indicating excellent model performance. Our results also showed that sentiment analysis features did not improve the model's performance, and word embeddings provided the best results in terms of feature extraction.

**Software and Tools:**

We used Python 3.8 for data processing, feature extraction, and model training. We used the following libraries: pandas, numpy, scikit-learn, nltk, and tensorflow. We also used pre-trained word embeddings from the GloVe library.

**Computing Environment:**

We performed all experiments on a computer with an Intel Core i7-9700K CPU, 16GB RAM, and a NVIDIA GeForce RTX 2060 GPU. We used the Anaconda distribution to manage our Python environment.

**Limitations:**

Our study has several limitations. First, our dataset may not be representative of all types of fake news. Second, our feature extraction techniques may not capture all important characteristics of the text data. Third, we used a limited set of machine learning algorithms, and other algorithms may perform better. Finally, our study did not investigate the impact of other factors, such as the source of the news article, on fake news detection.