Report Table:

	Base	**Col 1**	**Col 2**	**Col 3**	**Col 5**	**Col 6**	**Col 7**
Naive Bayes F1	0.3768	0.0035	0.0041	0.0013	0.0013	0.0025	0.0045
Naive Bayes Precision	0.3768	0.0035	0.0041	0.0013	0.0013	0.0025	0.0045
Naive Bayes Recall	0.3768	0.0035	0.0041	0.0013	0.0013	0.0025	0.0045
Decision Tree F1	0.6432	0.0987	0.0922	0.1218	0.1510	0.0955	0.0909
Decision Tree Precision	0.6432	0.0987	0.0922	0.1218	0.1510	0.0955	0.0909
Decision Tree Recall	0.6432	0.0987	0.0922	0.1218	0.1510	0.0955	0.0909
Random Forest F1	0.6716	-0.0042	-0.0054	0.0250	0.0176	-0.0100	-0.0054
Random Forest Precision	0.6716	-0.0042	-0.0054	0.0250	0.0176	-0.0100	-0.0054
Random Forest Recall	0.6716	-0.0042	-0.0054	0.0250	0.0176	-0.0100	-0.0054

Report Discussion Questions:

1. **Which is the best machine learning algorithm (classifier) for this task (for both the baseline and the improved classifier)? You need to discuss this per metric used to compute the performance.**

For the baseline classifier, the Random Forest classifier emerges as the standout performer with an F1 score of 0.6716, showcasing its superior overall performance compared to the Decision Tree and Naive Bayes classifiers. This score signifies a harmonious balance between precision and recall, indicating Random Forest's ability to accurately predict positive instances while effectively minimizing both false positives and false negatives within the dataset. Conversely, the Decision Tree classifier, although slightly less effective with an F1 score of 0.6432, still outperforms the Naive Bayes classifier, which records the lowest F1 score of 0.3768. This hierarchy of performance highlights the Random Forest classifier's reliability and suitability for the task based on the baseline evaluation.

With the improved classifier assessment, there are more discrepancies particularly in the Random Forest classifier's results, where negative values appear in certain columns, contrary to expected positive values for F1, precision, and recall scores. On the other hand, the Decision Tree and Naive Bayes classifiers exhibit varying performances across different features, with positive F1 scores indicating each feature's unique contribution to the model's performance.

2. **Which attributes contributed the most to each of the performance metrics for your improved model? Which contributed the least? (Write about this for each algorithm considered.)**

For the Naive Bayes classifier, 'Col 7' emerges as the most impactful feature based on its substantial F1 score of 0.0045 when excluded, signifying its predictive strength. Conversely, 'Col 1', 'Col 2', 'Col 3', and 'Col 5' exhibit minimal influence on the model, as indicated by their low F1 scores of 0.0013 or lower when omitted, implying their lesser importance or potential redundancy. In contrast, the Decision Tree classifier highlights 'Col 3' and 'Col 5' as highly influential features, given their significant impact on the model's predictive power when removed, with F1 scores of 0.1218 and 0.1510, respectively. Conversely, 'Col 1' and 'Col 2' show less contribution, as their exclusion leads to a smaller decrease in performance, suggesting they hold less discriminative information. The Random Forest classifier's results are inconclusive due to anomalous negative values. Assuming these are errors, 'Col 5' could be considered the most contributory feature, while 'Col 6' and 'Col 1' seem less influential based on their further negative F1 scores. Correcting these values would provide a clearer understanding of feature contributions. Overall, the feature set's impact varies significantly across algorithms, indicating potential for tailored feature engineering to enhance results. Methods like Mutual Information for Naive Bayes and capturing non-linear relationships for Decision Trees and Random Forests could improve model performance. Ensemble methods and alternative text representations like TF-IDF may also be beneficial, especially for text data present in the dataset.

3. **How good is your feature set for this task (for each algorithm)? (Base your response to this question to your answer from Question 2)**

Considering the substantial performance differences between the algorithm, the feature set seems more suited for Random Forest and Decision Tree than fro Naive Bayes. While Naive Bayes struggled consistently, showing poor performance even with feature adjustments and preprocessing adjustments, the performance was under 50%, Decision Trees and Random Forests exhibited relatively better performance. However, the improvement seen with these classifiers may be attributed more to their inherent strengths as models rather than the intrinsic quality of the feature set itself.

4. **If you had more time to work on this problem and do it more efficiently (in terms of performance), which features/text representation would you choose? Write 1-2 short paragraphs about the features sets you might want to try for this problem and why.**

If more time were available, exploring feature sets with interaction terms could be beneficial, as they capture relationships between features, enhancing the model's predictive capabilities. Incorporating domain-specific knowledge into the feature sets, such as using advanced text representations like Word2Vec or GloVe, could significantly enhance the model's understanding of textual data, enriching its semantic and syntactic understanding. Additionally, employing dimensionality reduction techniques like PCA or t-SNE on the features could improve algorithm efficiency, address the curse of dimensionality, and mitigate overfitting, leading to more accurate and efficient models.

Another area of exploration would involve experimenting with different tagsets, potentially including larger numbers of POS tags, to assess their impact on model performance. While a larger tagset might not necessarily improve classifier performance, comparing different tagsets could provide valuable insights into the optimal level of tag granularity for the task. Furthermore, refining text preprocessing methods could be valuable, as certain preprocessing steps like lowercase conversion or punctuation removal might inadvertently hinder classifier performance, especially in distinguishing proper nouns. Implementing a more nuanced text preprocessing model could enhance classifier understanding while ensuring essential distinctions like capitalization are preserved, albeit requiring careful design and implementation considerations.