Assignment 8 Report

Questions specific to this assignment:

1. **Naive Bayes by hand:**

- Precision: 0.9226
- Recall: 0.9597
- F1: 0.9408

2. **TF-IDF by hand:**

- Precision: 0.9773
- Recall: 0.8658
- F1: 0.9181

3. **Scikit-learn Implementations:**

   a) Naive Bayes:

   - Precision: 1.0000
   - Recall: 0.9396
   - F1: 0.9689

   b) TF-IDF Vectorizer + Logistic Regression (Default):

   - Precision: 1.0000
   - Recall: 0.8121
   - F1: 0.8963

4. **BERT:**

- Precision: 1.0000
- Recall: 0.9638
- F1: 0.8261

Experiment with one of the parameters in the Tfidf vectorizer (character ngram analyzer, stop words, norm, etc.). Briefly explain what the parameter is, and show how it affected the scores when you changed it.

We tested character n-grams with ngram_range=(2,3), and this parameter analyzes text as sequences of 2-3 characters instead of whole words.

Results comparison: Default word-level TF-IDF: F1: 0.8963,   Vocabulary size: 7,701

Character n-grams (2-3): F1: 0.9164 (improved), Vocabulary size: 14,908

The performance was improved by Character n-grams (capturing sub-word patterns), and we can find that the performance is more robust to misspellings. The larger vocabulary size indicates it found more unique character sequences useful for classification. Character n-grams splits text into 2-3 character sequences instead of whole words. This suggests that character-level features are valuable indicators for spam detection.

How long did this assignment take you? (1 sentence)

This assignment took me 4 days to complete.

Whom did you work with, and how? (1 sentence each)

I worked independently, using code examples from Rasika's lectures and online resources.

Which resources did you use? (1 sentence each)

geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/: Helped understand TF-IDF implementation.

Rasika's lecture materials: Provided foundation for BERT implementation.

A few sentences about:

What was the most difficult part of the assignment?

Implementing BERT with cross-validation and early stopping.

Debugging the TF-IDF implementation by hand.

What was the most rewarding part of the assignment?

Successfully implementing different text classification methods .

Seeing performance improvements with character n-grams.

Understanding how different approaches handle text classification.

What did you learn doing the assignment?

Importance of proper data handling and model evaluation.

Trade-offs between different text classification approaches.

Constructive and actionable suggestions for improving assignments, office hours, and class time are always welcome.

More guidance on BERT implementation details, especially for hyperparameter tuning and cross-validation setup.

Additional examples of text preprocessing techniques for the hand-implemented classifiers.