data preprocessing

The dataset is labeled as spam or ham

- Lowercasing: All messages are converted to lowercase for consistency.

- Punctuation and symbol removal: All non-alphanumeric characters are removed.

- Tokenization: Messages are split into individual words.

method

- Bag of Words (BoW) – creates vectors based on raw word frequency.

- TF-IDF (Term Frequency-Inverse Document Frequency) – adjusts word frequency by how common a word is across all documents.

experiment design

- The data was split into training (80%) and testing (20%) using train_test_split.

- Features were extracted using either BoW or TF-IDF.

hyper-parameters

alpha = 1.0 (Laplace smoothing; default value)

evaluation metric

- Accuracy – Proportion of correctly predicted messages.

- Precision, Recall, F1-score – Especially important for the spam class, since false positives can be problematic in real-world filtering

result

```
Accuracy for TF-IDF: 0.852017937219731
Classification Report for TF-IDF:
              precision    recall  f1-score   support

           0       0.85      1.00      0.92       950
           1       0.00      0.00      0.00       165

    accuracy                           0.85      1115
   macro avg       0.43      0.50      0.46      1115
weighted avg       0.73      0.85      0.78      1115
```

```
Accuracy for BoW: 0.9820627802690582
Classification Report for BoW:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       950
           1       0.98      0.90      0.94       165

    accuracy                           0.98      1115
   macro avg       0.98      0.95      0.96      1115
weighted avg       0.98      0.98      0.98      1115
```

findings

- BoW outperforms TF-IDF significantly in detecting spam messages.
- TF-IDF fails to detect spam (label 1) effectively, resulting in 0 precision and recall for that class.
- This suggests TF-IDF may have overly suppressed spam-indicative words due to the IDF weighting or insufficient representation in the training data.

Improvements

Use TfidfVectorizer from sklearn instead of manual implementation for more accurate IDF calculations.

Add n-grams (e.g., bigrams) to capture more contextual patterns in spam messages.

Apply SMOTE or class-weighting to handle class imbalance (spam: 165, ham: 950).

Try alternative classifiers (e.g., Logistic Regression, SVM) or ensemble methods for potentially better performance.

Feature Selection: Remove stop words or use chi-squared feature selection to retain discriminative terms.