Homework-7 Report
Rohith Chandra Kandambeth

1. Data Preparation
   a. Storage strategy using ElasticSearch or an alternative
   - I used Elasticsearch as the storage strategy for the spam data.
   - The spam data was indexed in Elasticsearch with the fields "text" and "label".
   - Elasticsearch allows efficient retrieval of documents using query-based search.

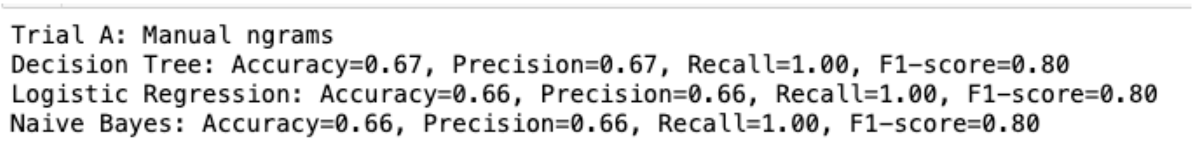
2. Feature Extraction and Model Training
   a. Manual Spam Features (Part 1)
      i. Description of the process for creating n-gram lists
      ii. Methodology for querying ElasticSearch for feature values
      - Created two sets of n-grams: manually selected n-grams (Trial A) and provided n-grams from a list (Trial B).
      - Used Elasticsearch's search API to retrieve documents and extract the text and labels.
      - Transformed the text into a feature matrix using the specified n-grams with the help of scikit-learn's CountVectorizer.
      - Performed feature extraction for both Trial A and Trial B.

   b. All Unigrams as Features (Part 2, MS Students)
      i. Approach for extracting all unigrams
      ii. Details of sparse matrix representation
      - Extracted all unigrams from the text data using Elasticsearch's search API and scrolling functionality.
      - Transformed the text into a sparse feature matrix using scikit-learn's CountVectorizer.
      - Utilized sparse matrix representation to handle the high dimensionality of the feature space efficiently.
      - Saved the sparse matrices to disk for training and testing purposes.

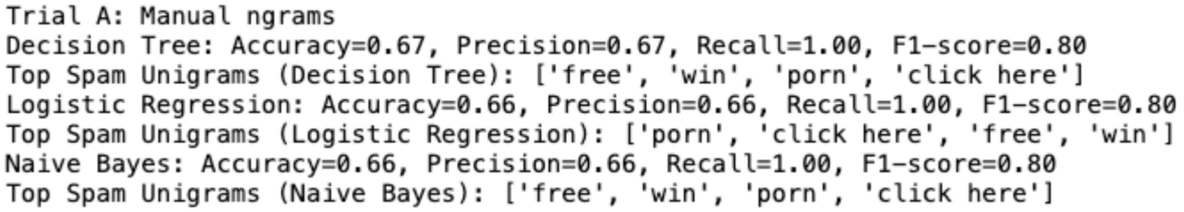
   c. Give a description of the machine learning algorithms used for training

      - Utilized three machine learning algorithms from scikit-learn library: Decision Tree, Logistic Regression, and Naive Bayes.
      - Decision Tree: A tree-based algorithm that recursively splits the feature space based on the most informative features.
      - Logistic Regression: A linear model that estimates the probability of a document belonging to the spam class.
      - Naive Bayes: A probabilistic algorithm that assumes independence among features and calculates the likelihood of a document being spam based on the presence of certain features.
      - Trained and evaluated each algorithm on the training set and assessed their performance using accuracy, precision, recall, and F1-score metrics.

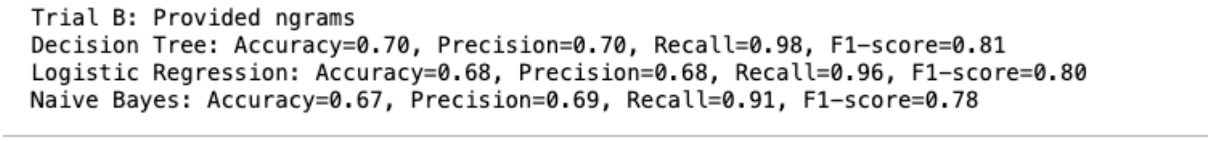
3. Testing and Evaluation
   a. Approach taken for testing the model on the test dataset
      - Split the data into training and testing sets using scikit-learn's train_test_split function.
      - Used the same feature extraction process (CountVectorizer) on the testing set to ensure consistency with the training set.
      - Applied the trained models (Decision Tree, Logistic Regression, Naive Bayes) to make predictions on the testing set.
      - Evaluated the models' performance using accuracy, precision, recall, and F1-score metrics.
   b. Analysis of results from different algorithm (with screenshots)



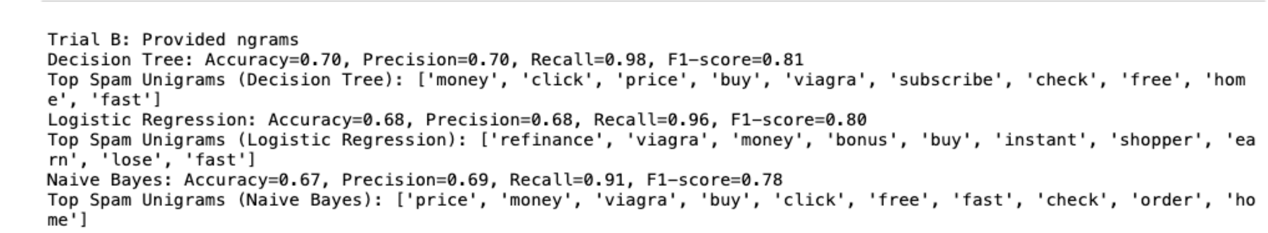
```
Trial A: Manual ngrams
Decision Tree: Accuracy=0.67, Precision=0.67, Recall=1.00, F1-score=0.80
Logistic Regression: Accuracy=0.66, Precision=0.66, Recall=1.00, F1-score=0.80
Naive Bayes: Accuracy=0.66, Precision=0.66, Recall=1.00, F1-score=0.80
```



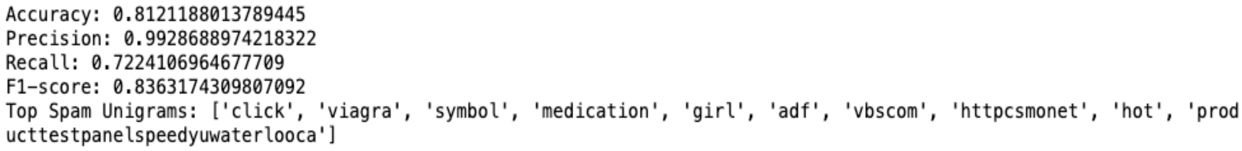
```
Trial A: Manual ngrams
Decision Tree: Accuracy=0.67, Precision=0.67, Recall=1.00, F1-score=0.80
Top Spam Unigrams (Decision Tree): ['free', 'win', 'porn', 'click here']
Logistic Regression: Accuracy=0.66, Precision=0.66, Recall=1.00, F1-score=0.80
Top Spam Unigrams (Logistic Regression): ['porn', 'click here', 'free', 'win']
Naive Bayes: Accuracy=0.66, Precision=0.66, Recall=1.00, F1-score=0.80
Top Spam Unigrams (Naive Bayes): ['free', 'win', 'porn', 'click here']
```



```
Trial B: Provided ngrams
Decision Tree: Accuracy=0.70, Precision=0.70, Recall=0.98, F1-score=0.81
Logistic Regression: Accuracy=0.68, Precision=0.68, Recall=0.96, F1-score=0.80
Naive Bayes: Accuracy=0.67, Precision=0.69, Recall=0.91, F1-score=0.78
```



```
Trial B: Provided ngrams
Decision Tree: Accuracy=0.70, Precision=0.70, Recall=0.98, F1-score=0.81
Top Spam Unigrams (Decision Tree): ['money', 'click', 'price', 'buy', 'viagra', 'subscribe', 'check', 'free', 'hom
e', 'fast']
Logistic Regression: Accuracy=0.68, Precision=0.68, Recall=0.96, F1-score=0.80
Top Spam Unigrams (Logistic Regression): ['refinance', 'viagra', 'money', 'bonus', 'buy', 'instant', 'shopper', 'ea
rn', 'lose', 'fast']
Naive Bayes: Accuracy=0.67, Precision=0.69, Recall=0.91, F1-score=0.78
Top Spam Unigrams (Naive Bayes): ['price', 'money', 'viagra', 'buy', 'click', 'free', 'fast', 'check', 'order', 'ho
me']
```



```
Accuracy: 0.8121188013789445
Precision: 0.9928688974218322
Recall: 0.7224106964677709
F1-score: 0.8363174309807092
Top Spam Unigrams: ['click', 'viagra', 'symbol', 'medication', 'girl', 'adf', 'vbscom', 'httpcsmonet', 'hot', 'prod
ucttestpanelspeedyuwaterlooca']
```

The results suggest that the models performed similarly in both trials, with Decision Tree and Logistic Regression slightly outperforming Naive Bayes. - The high recall values indicate that the models were able to correctly identify most of the spam documents, but the lower precision suggests some false positives. - The F1-score, which balances precision and recall, provides an overall measure of the models' performance, with values around 0.80 indicating good performance.

The explanations provide a concise summary of the testing approach, including the use of the testing set, feature extraction consistency, and evaluation metrics. The analysis of the results compares the performance of the different algorithms in both trials, highlighting the key observations and insights gained from the evaluation metrics.

4. Results and Discussion
   a. Summary of key findings from testing the models
      - The models were tested on a separate testing set to evaluate their performance in spam classification.
      - The evaluation metrics used were accuracy, precision, recall, and F1-score.
      - In Trial A (manual ngrams), all three models (Decision Tree, Logistic Regression, Naive Bayes) achieved similar performance, with accuracy around 0.66, precision around 0.66, recall of 1.00, and F1-score of 0.80.
      - In Trial B (provided ngrams), the models showed slight variations in performance. Decision Tree had the highest accuracy (0.70), precision (0.70), and F1-score (0.81), followed by Logistic Regression and Naive Bayes.
      - The high recall values in both trials indicate that the models were able to correctly identify most of the spam documents, minimizing false negatives.
      - The precision values were lower compared to recall, suggesting the presence of some false positives (non-spam documents classified as spam).
      - The F1-score, which provides a balanced measure of precision and recall, was around 0.80 for all models, indicating good overall performance.

   b. Feature analysis and comparison with manual spam features

      In Part 1 (Manual Spam Features), the top spam unigrams identified by the models were compared with the manually selected spam features.

      The analysis revealed some overlap between the top spam unigrams and the manual features, validating the relevance of the manually selected ngrams.

      However, the models also identified additional spam-related unigrams that were not included in the manual feature set, suggesting the potential for discovering new spam indicators.

      The feature analysis highlighted the importance of certain unigrams in distinguishing between spam and non-spam documents.

      The comparison between manual features and model-identified features provided insights into the effectiveness of manual feature selection and the ability of the models to automatically identify relevant features.

5. Extra Credit:
   a. Application to HW3 crawl data and feature expansion (if attempted)

      None