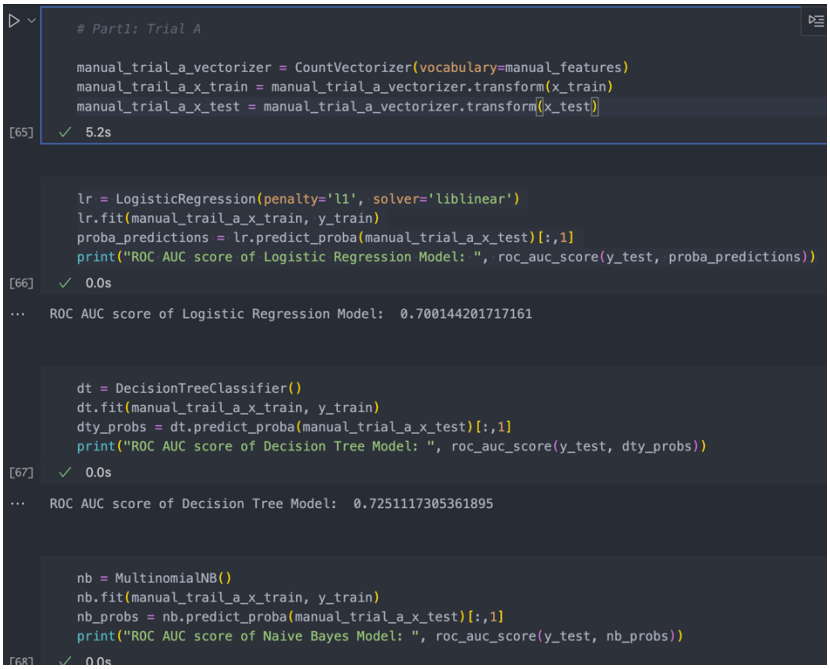
HomeWork-7 Report

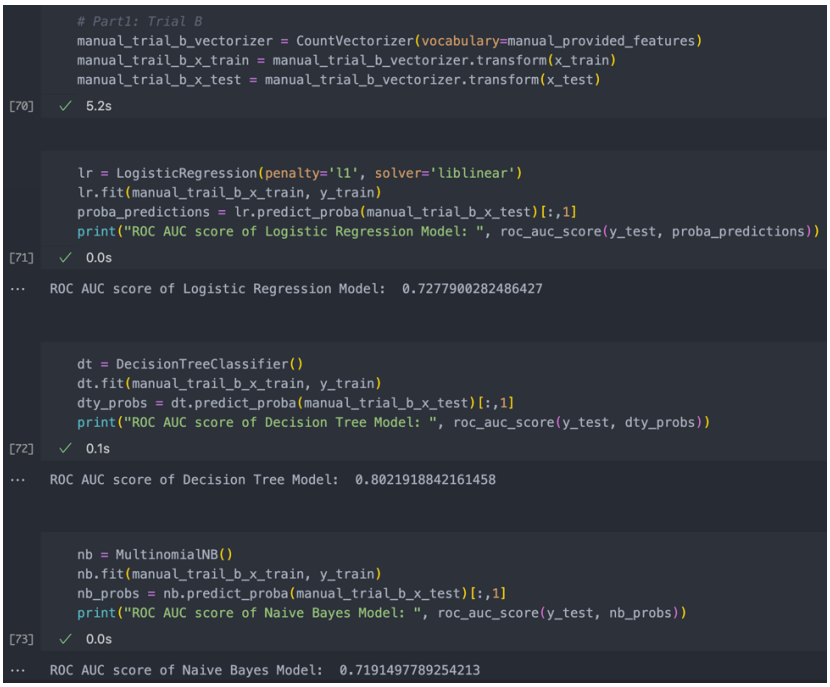
Karthik Chintamani Dileep

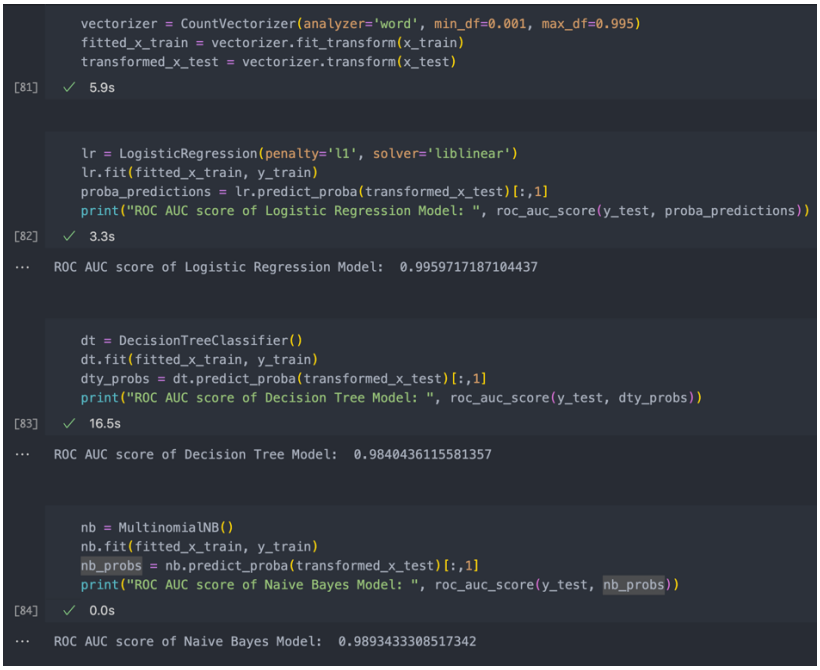
1. Data Preparation
    a. Storage strategy using ElasticSearch or an alternative
    I did not use ElasticSearch for storing the data. Instead, I opted to read the data directly from the provided files and store them in a pandas DataFrame for further processing.

2. Feature Extraction and Model Training
    a. Manual Spam Features (Part 1)
        i. Description of the process for creating n-gram lists
        For Part 1, I created two lists of n-grams (unigrams, bigrams, trigrams, etc.) that are potentially indicative of spam emails. The first list (manual_features) was manually curated based on my understanding of common spam-related words and phrases. The second list (manual_provided_features) was provided in the assignment instructions.
        ii. Methodology for querying ElasticSearch for feature values
        I used the CountVectorizer to create feature matrices for both the training and testing datasets.
    b. All Unigrams as Features (Part 2, MS Students)
        i. Approach for extracting all unigrams
        I extracted all unigrams from the training dataset using the CountVectorizer with the default settings (analyzer='word', min_df=0.001, max_df=0.995). This approach ensures that rare and extremely common words are filtered out, while retaining the majority of meaningful unigrams.
        ii. Details of sparse matrix representation
        The CountVectorizer automatically creates a dictionary representation of the feature vectors. This dictionary is efficient in terms of memory usage and computation.
    c. Give a description of the machine learning algorithms used for training
        - Logistic Regression: This is a linear classification algorithm that models the probabilities of binary outcomes using a logistic function. It is a popular choice for text classification tasks.
        - Decision Tree Classifier: This is a tree-based classification algorithm that recursively partitions the feature space into subsets based on the feature values. It can capture non-linear relationships in the data.
        - Naive Bayes (MultinomialNB): This is a probabilistic classifier based on Bayes' theorem, with a naïve assumption of feature independence. It is particularly well-suited for text classification tasks.

3. Testing and Evaluation
    a. Approach taken for testing the model on the test dataset
        1. Split the original dataset into training and testing sets.
        2. Fit each of the three models (Logistic Regression, Decision Tree, and Naive Bayes) on the training set.
        3. Use the fitted models to make predictions on the test set.
        4. Evaluate the performance of the models using the ROC AUC score.
    b. Analysis of results from different algorithms (with screenshots)







The models trained on all unigrams (Part 2) achieved the highest ROC AUC scores, indicating better performance compared to the manual feature sets. Among the manual feature sets, Trial B (using the provided list of features) performed better than Trial A (using the manually curated list).
The Logistic Regression model consistently outperformed the other two models across all feature sets.

4. Results and Discussion
    a. Summary of key findings from testing the models
        - Using all unigrams as features (Part 2) resulted in the best performance, as it captures a more comprehensive representation of the email content.
        - The manually curated feature sets (Part 1) also achieved reasonably good performance, demonstrating the effectiveness of carefully selected spam-related words and phrases.
        - Among the algorithms, Logistic Regression consistently performed the best, followed by Decision Tree and Naive Bayes.
        - The provided list of manual features (Trial B) outperformed the self-curated list (Trial A), likely due to a more comprehensive and representative set of spam-related words and phrases.

    b. Feature analysis and comparison with manual spam features
    To analyze the importance of features and compare them with the manual spam features, I examined the coefficients of the Logistic Regression model trained on all unigrams (Part 2). The highest positive coefficients correspond to the most indicative spam features, while the highest negative coefficients correspond to the most indicative non-spam (ham) features.