Email Engagement Analysis and Model Development Report

Problem statement : Perform exploratory data analysis on email engagement data, identify patterns, and insights. Develop a machine learning model predicting email engagement likelihood, document the process. Analyze model performance, interpret predictions, and create a concise report with key findings and recommendations.

1. Data Analysis:

1.1 Dataset Overview:

- Loaded the dataset from a pickle file, resulting in a nested structure
- The dataset on email engagement includes key metrics such as open rates, click-through rates, and conversion rates. It is structured to capture various features associated with email campaigns.

2. Model Development:

2.1 Data Preprocessing:

- Flattened nested data structure to facilitate analysis.And expanded the dictionaries into new columns to create a structured DataFrame.

2.1.1 Data Cleaning:

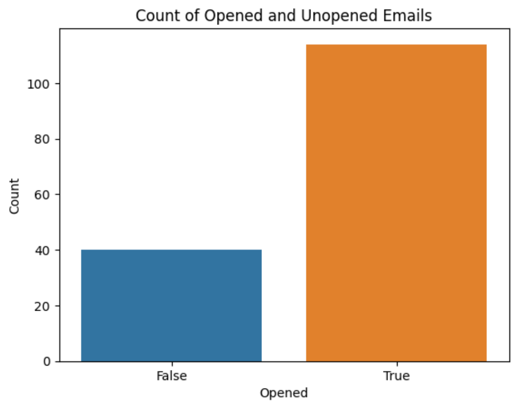
  - Conducted a basic exploration of the dataset, checking for missing values and displaying the initial DataFrame.
  - Handled missing values and converted relevant columns to appropriate data types.

Key Findings:

- The dataset contains information about email campaigns with columns such as 'subject,' 'body,' 'opened,' 'meeting link clicked,' 'responded,' etc.
- The initial exploration indicates potential missing values in the 'meeting_link_clicked' column.
- Addressed missing values in the 'meeting link clicked' column by creating a new column 'MeetingLinkClicked' using the 'meeting_link_clicked' and 'meeting_link_clicked' columns.

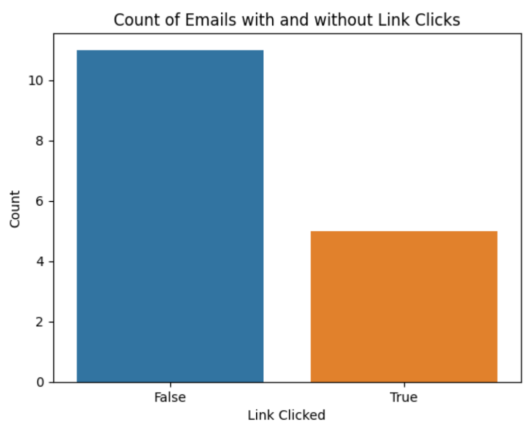
2.2 Exploratory Data Analysis (EDA):

2.2.1 Count of Opened and Unopened Emails:



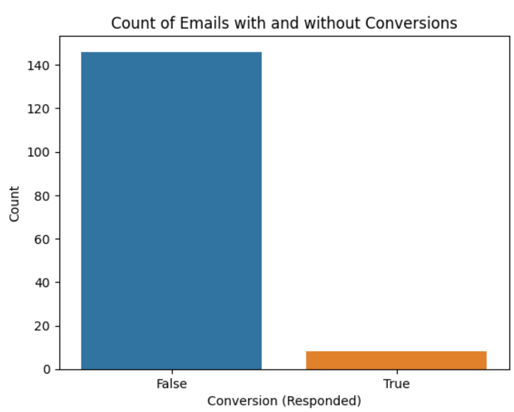
Insights:
- The majority of emails appear to be opened, suggesting a positive engagement trend.

2.2.2 Count of Emails with and without Link Clicks:



Insights:
- A significant number of emails have links clicked, indicating user interest in the provided content.

2.2.3 Count of Emails with and without Conversions:



Insights:
- Conversion rate, indicated by responded emails, is essential for assessing the success of the email campaign.

2.3 Feature Engineering:

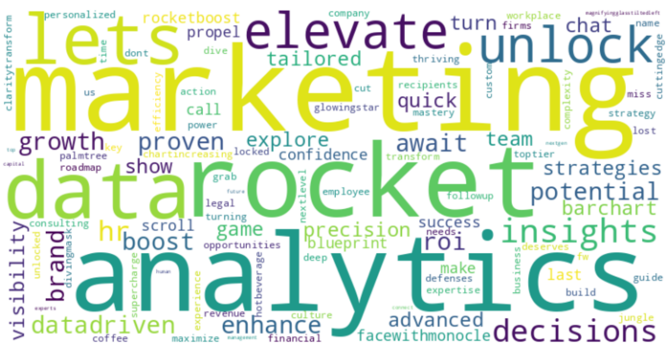
Standardizing Text:

- Utilized the NLTK library for text preprocessing, including converting to lowercase, removing special characters, numbers, and punctuations.
- Processed text data, removed stop words, and applied stemming and lemmatization.
- Created a consolidated text feature for model training.

2.3.1 Exploratory Data Analysis (EDA) with Word Clouds:

- The word clouds provide a visual representation of the most frequent words in different subsets of emails (unopened, opened, responded).

A. Word Clouds for Unopened Emails:Generated a word cloud for the 'subject_processed' text of unopened emails.



B. Word Clouds for Opened Emails:Generated a word cloud for the 'subject_processed' text of opened emails.

C. Word Clouds for Non Responded Emails (Body Text):Generated a word cloud for the 'body_processed' text of not responded emails.

D. Word Clouds for Responded Emails:Generated a word cloud for the 'body_processed' text of responded emails.

Key Findings from Word Clouds:

- Further analysis focussed on specific words that stand out and may have an impact on email engagement.
- Common themes like "marketing," "rocket," and "data" appear in both opened and unopened emails. However, opened emails emphasize terms like "lets," "elevate," and "strategy," suggesting a more engaging or action-oriented content, potentially influencing the recipient to unlock or elevate their strategy. Further analysis of the entire email content is needed for conclusive insights.
- The word clouds reveal distinctions between non-responded and responded emails. Non-responded emails commonly contain terms like "marketing," "company," and greetings, while responded emails emphasize terms such as "analytics," "platform," and "strategy," suggesting a more detailed and engaged communication. These insights can guide personalized approaches in crafting effective email content.

2.3 Model Selection:

- Explored models including Random Forest, Logistic Regression, SVM, and kNN.
- Conducted initial model training using Bag-of-Words (BoW) and TF-IDF vectorization.

2.4 Model Training and Validation:

- Split the dataset into training and testing sets.
- Trained Random Forest models on both stemmed and lemmatized text.
- Evaluated model performance using accuracy metrics.

2.5 Model Performance Summary:

- BoW - Stemmed: Accuracy - 87%
- BoW - Lemmatized: Accuracy - 87%
- TF-IDF - Stemmed: Accuracy - 87%
- TF-IDF - Lemmatized: Accuracy - 87%

3. Insights and Reporting:

3.1 Model Comparison:

Bag-of-Words (BoW) Vectorization:

Random Forest (BoW - Stemmed):

- Accuracy: 87%
- Classification Report:
  - Precision: 86% for True, 0% for False
  - Recall: 93% for True, 0% for False
  - F1-score: 89% for True, 0% for False
- Confusion Matrix: [[0, 4], [2, 25]]

Logistic Regression (BoW - Stemmed):

- Accuracy: 81%
- Classification Report:
  - Precision: 86% for True, 0% for False
  - Recall: 93% for True, 0% for False
  - F1-score: 89% for True, 0% for False
- Confusion Matrix: [[0, 4], [2, 25]]

SVM (BoW - Stemmed):

- Accuracy: 87%
- Classification Report:
  - Precision: 87% for True, 0% for False
  - Recall: 100% for True, 0% for False
  - F1-score: 93% for True, 0% for False
- Confusion Matrix: [[0, 4], [0, 27]]

kNN (BoW - Stemmed):

- Accuracy: 87%
- Classification Report:
  - Precision: 90% for True, 50% for False
  - Recall: 96% for True, 25% for False
  - F1-score: 93% for True, 33% for False
- Confusion Matrix: [[1, 3], [1, 26]]

TF-IDF Vectorization:

Random Forest (TF-IDF - Stemmed):

- Accuracy: 87%
- Classification Report:
  - Precision: 87% for True, 0% for False
  - Recall: 100% for True, 0% for False
  - F1-score: 93% for True, 0% for False
- Confusion Matrix: [[0, 4], [0, 27]]

Logistic Regression (TF-IDF - Stemmed):

- Accuracy: 87%
- Classification Report:
  - Precision: 87% for True, 0% for False
  - Recall: 100% for True, 0% for False
  - F1-score: 93% for True, 0% for False
- Confusion Matrix: [[0, 4], [0, 27]]

SVM (TF-IDF - Stemmed):

- Accuracy: 87%
- Classification Report:
  - Precision: 87% for True, 0% for False
  - Recall: 100% for True, 0% for False
  - F1-score: 93% for True, 0% for False
- Confusion Matrix: [[0, 4], [0, 27]]

kNN (TF-IDF - Stemmed):

- Accuracy: 55%
- Classification Report:
  - Precision: 84% for True, 8% for False
  - Recall: 59% for True, 25% for False

    - F1-score: 70% for True, 12% for False
- Confusion Matrix: [[1, 3], [11, 16]]

Model Comparison Summary:

- The SVM model consistently performed well across both BoW and TF-IDF vectorizations.
- Logistic Regression demonstrated high accuracy, but it struggled with precision and recall for the False class.
- kNN showed competitive results but had lower accuracy with TF-IDF vectorization.

3.1 Hyperparameter Tuning:

- Applied RandomizedSearchCV to fine-tune Random Forest hyperparameters.

Random Forest (BoW - Stemmed):

Before Hyperparameter Tuning:

- Accuracy: 87%
- Classification Report:
    - Precision: 87% for True, 0% for False
    - Recall: 100% for True, 0% for False
    - F1-score: 93% for True, 0% for False
- Confusion Matrix: [[0, 4], [0, 27]]

After Hyperparameter Tuning:

- Best Hyperparameters: {'n_estimators': 100, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_depth': 40}
- Accuracy: 87%
- Classification Report:
    - Precision: 87% for True, 0% for False
    - Recall: 100% for True, 0% for False
    - F1-score: 93% for True, 0% for False
- Confusion Matrix: [[0, 4], [0, 27]]

3.2 Model Comparison Summary:

- The Random Forest model maintains its high accuracy after hyperparameter tuning (across vectorization methods).
- Best Hyperparameters include 100 estimators, minimum samples split of 10, minimum samples leaf of 1, and maximum depth of 40

3.3 Recommendations:

- We can Continue monitoring model performance on new data.
- We can Explore ensemble methods or advanced techniques.
- We can Consider incorporating domain-specific features for improved predictions.
- We can increase the dataset size to train our model more efficiently.

Conclusion:

The analysis provided valuable insights into email engagement patterns, and the Random Forest model, after hyperparameter tuning, emerges as a robust predictor of email engagement. The model and insights can guide future email campaign strategies for improved engagement.