Name: Kunal Rane

Email: kunalrane1397@gmail.com

Phone: 7276471513

EMAIL CLASSIFICATION ASSIGNMENT

TABLE OF CONTENTS

1. INTRODUCTION

Email communication is an integral part of modern business and personal interactions. The requirement for automatic email classification is critical as the volume of emails keeps increasing. **The goal of this Email Classification Code Assignment is to create an Artificial Intelligence (AI) model that will classify emails into category according to their content.**

In this documentation, the code and procedures used to do this task are covered. Data collection, category information decoding, data preparation, exploratory data analysis, model selection, training, and evaluation are all parts of the assignment. Data processing methods, model construction, assessment measures, outcomes, and insights are provided in this documentation. To determine which machine learning models are most effective at classifying emails, several are taken into consideration.

2. PROJECT FLOW

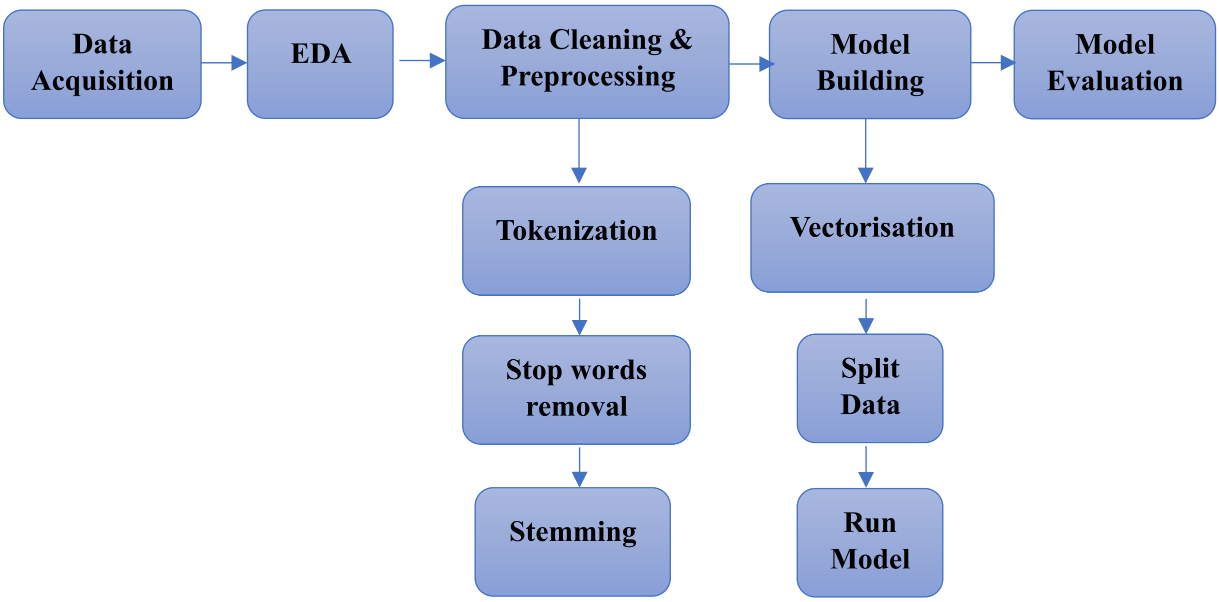


Fig. 1 Project flow of NLP

3. DATA ACQUISITION

a. Data files downloaded from the given resource. Multiple .cats file and text file were found. (https://bailando.berkeley.edu/enron_email.html)

b. The .cats files contain category information that needs to be associated with the email texts for classification purposes.

c. Using python data was extracted from the all files and folders and decoded to the categories correctly.

By successfully decoding the .cats files and mapping the numerical labels to categories, foundation for email classification is established. This process ensured that each email was associated with a meaningful category, enabling the subsequent steps of data preprocessing, model building, and evaluation to proceed effectively.

4. EXPLORATORY DATA ANALYSIS (EDA)

The Exploratory Data Analysis (EDA) conducted on the provided dataset revealed valuable insights into the email data's characteristics and distribution. Here is a summary of the key findings from the EDA process:

No of columns	2
No of records	1701

Table 1 Dataset information

Independent variable	X = Email text
Dependent variable	Y = Category

Table 2 Features of Interest

4.1 Category Distribution:

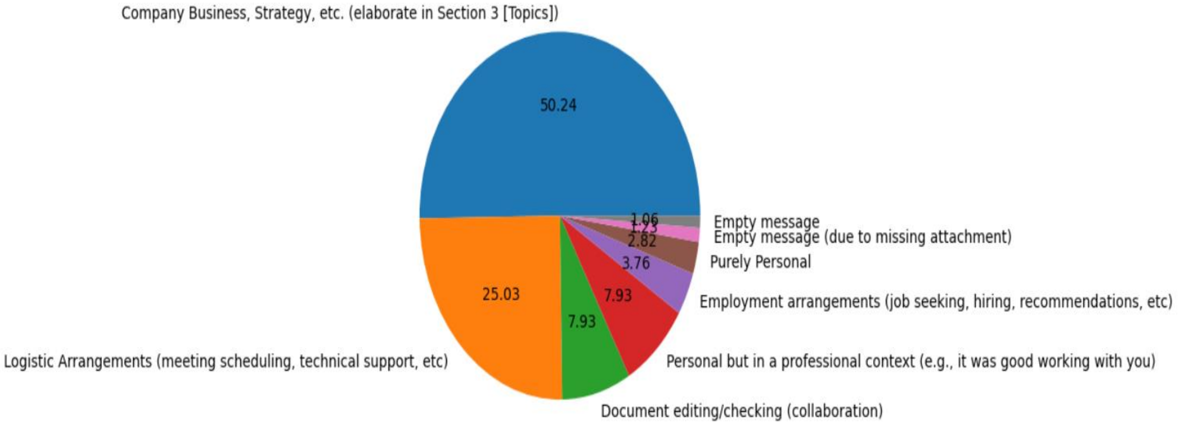


Fig. 2 Pie chart of categories

To understand the data distribution pie chart was plotted and the different categories along with their proportions was understood.

The distribution of these categories is as follows:

Category	**Value count**
Company Business, Strategy, etc	855 emails
Logistic Arrangements	426 emails
Document editing/checking	135 emails
Personal but in a professional context	135 emails
Employment arrangements	64 emails
Purely Personal	48 emails
Empty message (due to missing attachment)	21 emails
Empty message	18 emails

Table 3 Distribution of Categories

4.2 Text Characteristics:

- Several characteristics of the email text were analyzed, including:
  - Number of characters in each email.
  - Number of words in each email.
  - Number of sentences in each email.
- Descriptive statistics for these characteristics were computed for the entire dataset:

Mean number of characters	7440.46
Mean number of words	1389.54
Mean number of sentences	40.23

Table 4 Distribution of Categories

4.3 Histograms:

Histograms were plotted to visualize the distribution of the number of characters and words in emails across different categories. The histograms provided insights into the variations in text length among categories.

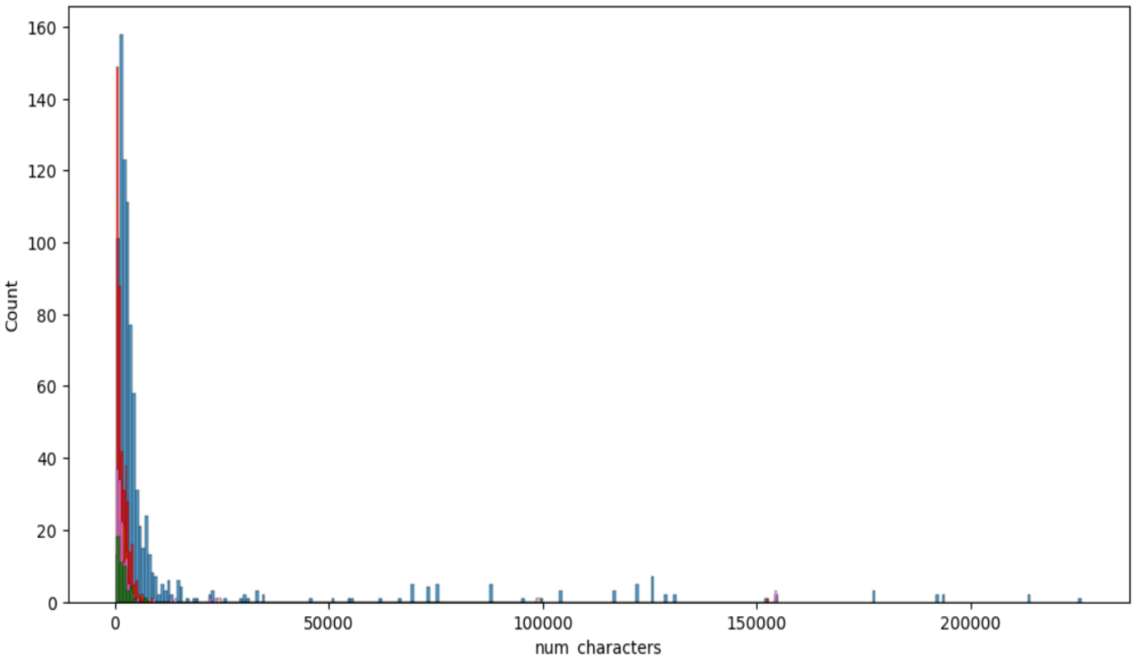


Fig. 3 Count vs Num_characters

Fig. 3 count vs num_chacracters shows if the emails histograms shows that the Company Business category email generally are of more characters compared to other categories. It is followed by Logistic Arrangements, Document editing/checking and Personal but in a professional context category.

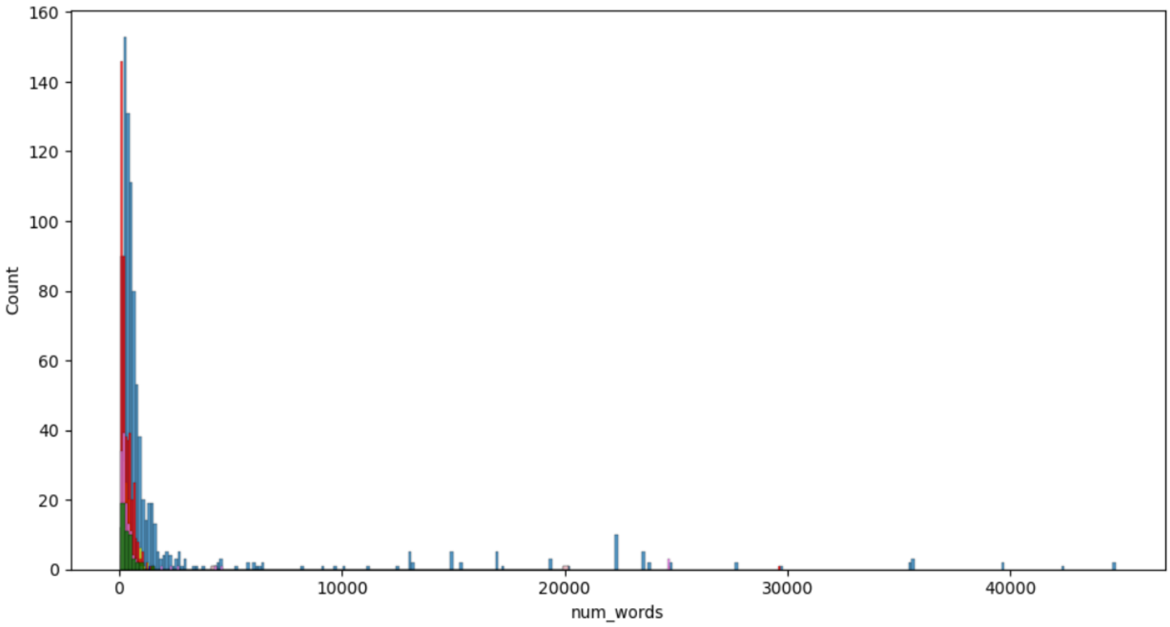
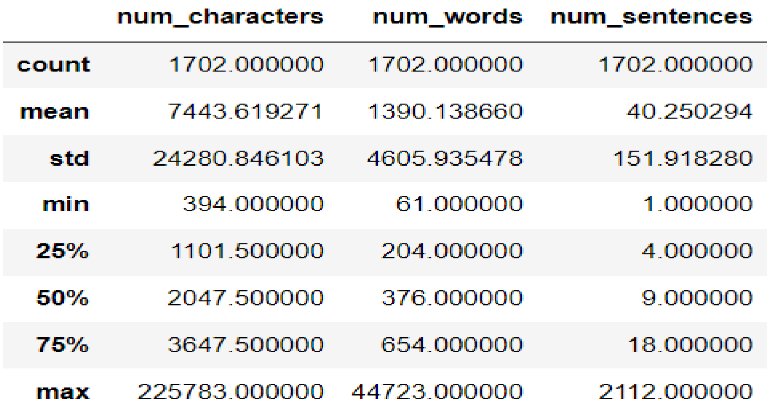


Fig. 4 Count vs Num_words

Fig. 3 count vs num_words shows similar trend like the num_characters histogram. Emails in the Company Business category typically have more characters than emails in other categories. Logistic Arrangements, document editing and checking, and Personal but in a professional environment are the categories that come after it.



	num_characters	num_words	num_sentences
count	1702.000000	1702.000000	1702.000000
mean	7443.619271	1390.138660	40.250294
std	24280.846103	4605.935478	151.918280
min	394.000000	61.000000	1.000000
25%	1101.500000	204.000000	4.000000
50%	2047.500000	376.000000	9.000000
75%	3647.500000	654.000000	18.000000
max	225783.000000	44723.000000	2112.000000

Table 5 Description of the number of characters, words and sentence of email

4.4 Word Cloud Analysis:

- Word Clouds were generated to visually represent the most frequent words in the email text for two specific categories, namely 'Company Business, Strategy, etc.' and 'Logistic Arrangements.' These Word Clouds provide a qualitative view of the prominent words in these categories.
- For the 'Company Business, Strategy, etc.' category, a Word Cloud was created to illustrate the most common terms within emails falling into this category.
- Similarly, a Word Cloud was generated for the 'Logistic Arrangements' category to highlight the prevalent words in emails related to logistical matters.

- ➢ Word Clouds are useful for identifying recurring themes and keywords within specific categories, aiding in the understanding of the content distribution within these categories.

These Word Cloud visualizations offer an additional layer of insight into the textual content of emails within the selected categories, complementing the quantitative analysis conducted during the EDA process.

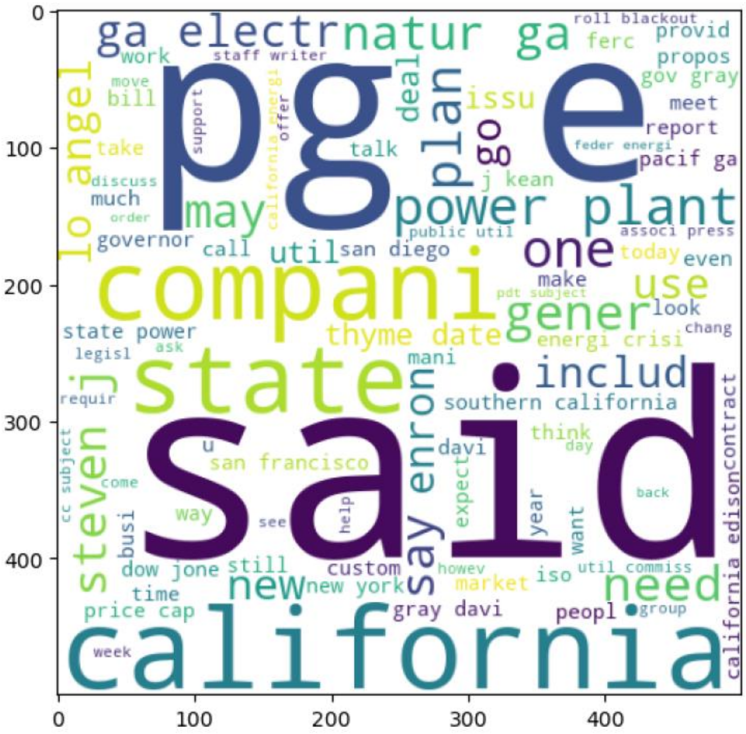


Fig. 5 Word Cloud for the Company Business category

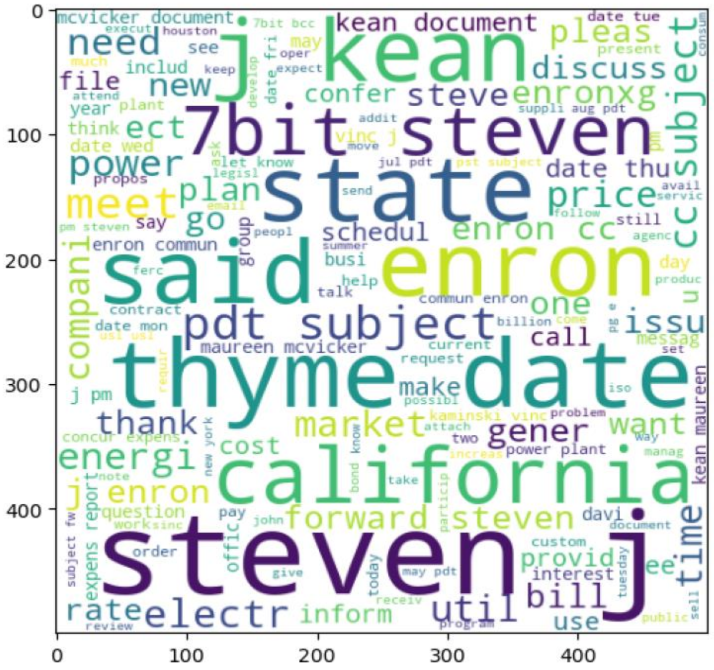
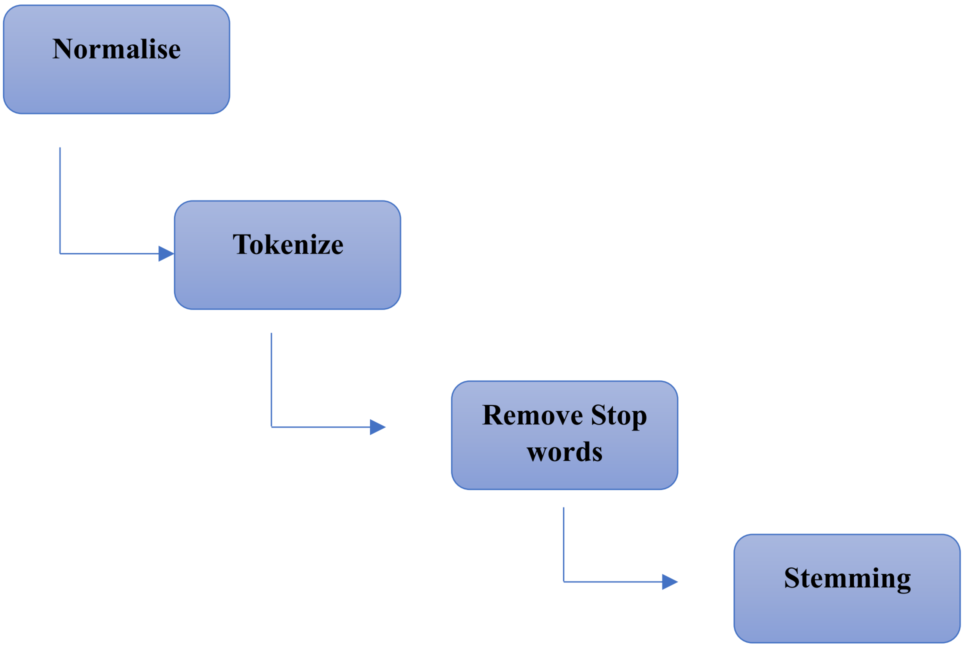


Fig. 6 Word Cloud for the Logistic Arrangements category

5. DATA CLEANING AND PREPROCESSING

- Removing the numbers, symbols, punctuations, spaces, special characters from the text data.
- Normalizing the data by lowering the case of the text.
- Tokenising the data from each sentence of the emails.
- Stemming the words.
- Removing the stop words from the text data.
- Rejoin meaningful stem words in single string like a sentence.
- Finally vectorizing all the words into vector matrixes


Normalise

Tokenize

Remove Stop words

Stemming

Fig. 7 Flow of data cleaning and preprocessing

6. MODEL BUILDING AND TRAINING

6.1 Data Split

The data was split in 80% training set and 20% test set

6.2 Modelling Methods:

1) Logistic Regression:

- Logistic Regression is a simple yet effective classification algorithm.
- In our analysis, combination of CountVectorizer and TfidfTransformer to convert the text data into numerical features is used.
- The Logistic Regression model was trained on the preprocessed email text.

2) **Naive Bayes Classifier:**

- The Naive Bayes Classifier is a probabilistic algorithm that is particularly suited for text classification tasks.
- Similar to Logistic Regression, CountVectorizer and TfidfTransformer for feature extraction is used.
- The MultinomialNB variant of Naive Bayes was employed.

3) **XGBoost Classifier:**

- XGBoost is an ensemble learning method known for its high performance.
- CountVectorizer and TfidfTransformer were used to prepare the data for training.
- The XGBoost Classifier was chosen for its robustness and predictive power.

4) **Random Forest Classifier:**

- Random Forest is another ensemble learning technique based on decision trees.
- CountVectorizer and TfidfTransformer are utilized to transform the text data.
- Random Forest Classifier was employed for its ability to handle complex data and reduce overfitting.

7. RESULTS AND CONCLUSION

Before addressing the results, it's important to note that the dataset initially suffered from class imbalance issues. To address this, the Synthetic Minority Over-sampling Technique for Extremely Imbalanced Data (SMOTE-ENN) was applied to balance the dataset.

Here are the accuracy results before and after balancing the dataset:

Machine learning Model	**Before Balancing**	**After Balancing:**
Logistic Regression	65%	95%
Naive Bayes Classifier	62%	82%
Random Forest Classifier	62%	98%
XGBoost Classifier	69%	99%

Table 6 Comparison of ML models

Analysis of Results:

1) **Logistic Regression:**

Logistic Regression showed significant improvement in accuracy after balancing the dataset, indicating its effectiveness in classification tasks when data balance is achieved.

2) **Naive Bayes Classifier:**

The Naive Bayes Classifier also benefited from dataset balancing but to a lesser extent compared to Logistic Regression and XGBoost.

3) **Random Forest Classifier:**

Similar to XGBoost, Random Forest exhibited notable accuracy both before and after balancing the dataset, indicating its robustness.

4) **XGBoost Classifier:**

XGBoost demonstrated the highest accuracy among all models, even before dataset balancing. After balancing, it achieved near-perfect accuracy, showcasing its ability to handle complex classification problems.

Conclusion

- The dataset initially suffered from class imbalance, which affected the performance of the models, especially Logistic Regression and Naive Bayes.
- Balancing the dataset significantly improved the accuracy of all models.
- XGBoost outperformed other models both before and after balancing, achieving an impressive accuracy of 99% after balancing.
- Random Forest also showed excellent performance with 98% accuracy post-balancing.
- The choice of balancing technique had a substantial impact on model performance, highlighting the importance of handling class imbalance effectively.
- The high accuracy scores indicate that the models are effective in categorizing emails into their respective categories based on content.

Recommendations:

Based on the results, the XGBoost classifier is recommended as it achieved the highest accuracy. It should be deployed for categorizing emails into different content categories.

8. Challenges faced?

1. Data received is of raw format
2. Highly Imbalanced dataset.

How did I overcome them

1. Data is pre-processed and cleaned
2. Dataset is balanced applying different SMOTEENN technique to handle imbalance

9. CODE OVERVIEW

(https://github.com/kunalrane13/Email-Classification-Using-NLP/blob/main/Email%20Clssification%20using%20NLP.pdf**)**