Problem statement: Building a smarter AI-POWERED Spam Classifier

problem definition

1.Background

The increasing volume of digital communication, especially through email and messaging platforms, has led to a surge in spam messages. Spam not only clogs inboxes but can also be a vector for various forms of cyberattacks, phishing attempts, and the spread of malware. Traditional spam filters often fall short in accurately detecting and classifying spam due to the evolving tactics employed by spammers. To address this challenge, there is a need to develop a smarter Al-powered spam classifier that can reliably identify and filter out spam messages while minimizing false positives.

2.Problem Statement:

The goal is to design and develop an advanced Al-powered spam classifier capable of accurately distinguishing between legitimate messages and spam across various digital communication channels, including email, messaging apps, and social media platforms.

3 Key-Objectives:

a. High Accuracy: Achieve a high level of accuracy in spam detection to minimize the chances of false negatives and false positives.

b. Adaptability: Create a classifier that can adapt to evolving spam tactics and handle different types of spam, including text, image, and multimedia-based spam.

c. Efficiency: Ensure that the classifier operates efficiently and does not significantly impact the performance or latency of the communication platform.

d. User-Friendly: Develop a user-friendly interface or integration that allows users to customize and fine-tune spam filtering preferences.

4. Data Collection:

Gather a diverse and extensive dataset of both spam and legitimate messages. The dataset should encompass various languages, message formats, and types of spam, such as phishing, advertising, and malicious attachments.

5. Model Development:

Create a machine learning or deep learning model that leverages natural language processing (N LP) and computer vision techniques to analyze and classify incoming messages. This model should be capable of learning from the dataset and continuously improving its accuracy over time.

6.Evaluation Metrics:

Define appropriate evaluation metrics, such as precision, recall, Fl score, and accuracy, to assess the classifier's performance. These metrics will help measure how well the model identifies spam while avoiding false positives.

7. Integration:

Develop a seamless integration process for deploying the Al-powered spam classifier into various digital communication platforms, including email clients, messaging apps, and social media.

8.Feedback Loop:

Implement a feedback mechanism that allows users to report false positives and false negatives, enabling continuous model refinement.

9.Security and Privacy:

Ensure that the classifier complies with data privacy regulations and does not compromise user privacy during the spam detection process.

10. Scalability:

Design the system to be scalable, capable of handling high volumes of messages in real-time, as the user base grows.

Design thinking

Design thinking is a human-centered approach to problem-solving that focuses on understanding the user's needs and creating innovative solutions. When applying design thinking to the development of a smarter Al-powered spam classifier, the following stages and principles should be considered:

1.Empathize: Understand User Needs and Pain Points

Conduct user research and surveys to gather insights into the challenges users face with spam messages.

Interview users to understand their preferences for spam filtering and their tolerance for false positives.

Analyze existing spam filtering solutions to identify common pain points and shortcomings.

2.Define: Clearly Define the Problem and Objectives

Define the specific goals and objectives of the Al-powered spam classifier, considering factors such as accuracy, adaptability, and efficiency.

Create user personas to represent different types of users who will interact with the classifier.

Develop a clear problem statement that outlines the key challenges and desired outcomes.

3.Ideate: Brainstorm Innovative Solutions

Organize brainstorming sessions with cross-functional teams to generate creative ideas for spam detection and classification.

Encourage diverse perspectives and approaches, such as leveraging machine learning, deep learning, and data augmentation techniques.

Prioritize ideas based on their feasibility, potential impact, and alignment with user needs.

4.Prototype:Build Prototypes and Mockups

Develop prototypes or mockups of the Al-powered spam classifier's user interface to visualize its functionality.

Create a proof-of-concept model using a subset of the data to test the viability of different machine learning algorithms.

Experiment with different feature engineering techniques and data preprocessing methods.

5.Test: Gather Feedback and Iterate

Conduct usability testing sessions with users to gather feedback on the prototype's design and functionality.

Evaluate the performance of the Al model using a representative dataset, measuring key metrics like precision, recall, and accuracy.

Iterate on the design and model based on user feedback and performance results.

6.Implement: Develop the AI-Powered Spam Classifier

Build the final AI model using the selected machine learning or deep learning algorithms.

Develop the user interface or integration for seamless deployment into digital communication platforms.

Ensure that the implementation complies with data privacy regulations and security best practices.

7. Monitor: Establish Ongoing Monitoring and Maintenance

Implement monitoring tools to track the classifier's performance in real-time.

Set up alerts for unusual patterns or issues that may arise during operation.

Continuously update and retrain the Al model to adapt to new spam tactics and improve accuracy.

8. Scale: Plan for Scalability and Growth

Design the system architecture to accommodate increased volumes of messages and users.

Consider cloud-based solutions for scalability and load balancing.

Prepare for expansion to different languages and communication platforms.

9. Communicate and Educate: Provide User Documentation and Training

Develop comprehensive user documentation and training materials to help users understand and utilize the spam classifier effectively.

Offer customer support channels for users to seek assistance and report issues.

10. Iterate: Encourage Continuous Improvement

Foster a culture of continuous improvement by regularly soliciting user feedback and conducting post-implementation reviews.

Allocate resources for research and development to stay ahead of emerging spam threats.

Innovation: Ensemble model

Abstract :

Dynamic Spam Detection using Ensemble models

Developing an ensemble that dynamically selects and combines di erent models based on the characteristics of incoming or message ,adopting to changing spam patterns.

brief description:

Ensemble methods can be employed effectively to detect spam in real time, such as when a user receives a new email or social media message. The idea behind using ensemble methods is to combine the predictions of multiple machine learning models to enhance classification accuracy and robustness. Here we are going to implement ensemble methods for real-time spam detection:

1. Data Collection and Preprocessing

*** Collect a diverse and labeled dataset of both spam and non-spam (ham) messages, representative of the types of messages your users typically receive.**

***Preprocess the data by cleaning and tokenizing the text, handling imbalanced classes, and extracting relevant features.**

2.Feature Engineering

*** Extract relevant features from the text data, such as word frequencies, character n-grams, or TF-IDF (Term Frequency-Inverse Document Frequency) values.**

***Consider incorporating additional features like sender information, email headers, and metadata for social media messages.**

3.Model Selection

***Choosing the effective machine learning models that are suitable for text classification which results in better classification.**

4.Ensemble Construction

***Create an ensemble of these diverse models.**

*** Bagging where Training multiple models on different subsets of the data and aggregate their predictions (e.g., Random Forest).**

5.Real-Time Integration

***Developing a real-time processing pipeline that integrates with social media platform.**

***When a new message arrives, extract relevant features and feed them into the ensemble of**

models.

***Aggregate the individual model predictions using the chosen ensemble technique includes bagging.**

6.Thresholding and Decision Rules

*** Set an appropriate threshold for the ensemble's output probabilities to determine whether a message is classified as spam or not.**

***Implement decision rules to handle borderline cases, where multiple models may not agree on the classification.**

7. Model Updating

***Periodically retrain the individual models within the ensemble to adapt to changing spam tactics and patterns.**

8.Performance Monitoring

*** Continuously monitor the performance of the ensemble in real-time to ensure that it meets accuracy and latency requirements.**

Ensemble methods are powerful for real-time spam detection because they can combine the strengths of multiple models, making the system more robust to various types of spam. However, it's essential to regularly update and maintain the ensemble to keep up with evolving spamming techniques and patterns.

BUILIDING THE MODEL AND DEVELOPING PREPROCESSING STEPS

STEPI:TOKENIZATION

The first step of our model is gathering the dataset as a sentence and breaking it into understandable parts (words) and printing the same for first few sentences of our dataset using head().

"AFTER TOKENIZATION: [['Subject', ':', 'enron', 'methanol', ';', 'meter', '#', ':', '988291', 'this', 'is', 'a', 'follow', 'up', 'to', 'the', 'note', 'i', 'gave', 'you', 'on', 'monday', ',', '4', '/', '3', '/', '00', '{', 'preliminary', 'flow', 'data', 'provided', 'by', 'daren', '}', '.', 'please', 'override', 'pop', \"'\", 's', 'daily', 'volume', '{', 'presently', 'zero', '}', 'to', 'reflect', 'daily', 'activity', 'you', 'can', 'obtain', 'from', 'gas', 'control', '.', 'this', 'change', 'is', 'needed', 'asap', 'for', 'economics', 'purposes', '.'], ['Subject', ':', 'hpl', 'nom', 'for', 'january', '9', ',', '2001', '(', 'see', 'attached', 'file', ':', 'hplnol', '09', '.', 'xls', ')', '-', 'hplnol', '09', '.', 'xls'], ['Subject', ':', 'neon', 'retreat', 'ho', 'ho', 'ho', ',', 'we', \"'\", 're', 'around', 'to', 'that', 'most', 'wonderful', 'time', 'of', 'the', 'year', '-', '-', '-', 'neon', 'leaders', 'retreat', 'time', '!', 'i', 'know', 'that', 'this', 'time', 'of', 'year', 'is', 'extremely', 'hectic', ',', 'and', 'that', 'it', \"'\", 's', 'tough', 'to', 'think', 'about', 'anything', 'past', 'the', 'holidays', ',', 'but', 'life', 'does', 'go', 'on', 'past', 'the', 'week', 'of', 'december', '25', 'through', 'january', '1', ',', 'and', 'that', \"'\", 's', 'what', 'i', \"'\", 'd', 'like', 'you', 'to', 'think', 'about', 'for', 'a', 'minute', '.', 'on', 'the', 'calender', 'that', 'i', 'handed', 'out', 'at', 'the', 'beginning', 'of', 'the', 'fall', 'semester', ',', 'the', 'retreat', 'was', 'scheduled', 'for', 'the', 'weekend', 'of', 'january', '5', '-', '6', '.', 'but', 'because', 'of', 'a', 'youth', 'ministers', 'conference', 'that', 'brad', 'and', 'dustin', 'are', 'connected', 'with', 'that', 'week', ',', 'we', \"'\", 're', 'going', 'to', 'change', 'the', 'date', 'to', 'the', 'following', 'weekend', ',', 'january', '12', '-', '13', '.', 'now', 'comes', 'the', 'part', 'you', 'need', 'to', 'think', 'about', '.', 'i', 'think', 'we', 'all', 'agree', 'that', 'it', \"'\", 's', 'important', 'for', 'us', 'to', 'get', 'together', 'and', 'have', 'some', 'time', 'to', 'recharge', 'our', 'batteries', 'before', 'we', 'get', 'to', 'far', 'into', 'the', 'spring', 'semester', ',', 'but', 'it', 'can', 'be', 'a', 'lot', 'of', 'trouble', 'and', 'difficult', 'for', 'us', 'to', 'get', 'away', 'without', 'kids', ',', 'etc', '.', 'so', ',', 'brad', 'came', 'up', 'with', 'a', 'potential', 'alternative', 'for', 'how', 'we', 'can', 'get', 'together', 'on', 'that', 'weekend', ',', 'and', 'then', 'you', 'can', 'let', 'me', 'know', 'which', 'you', 'prefer', '.', 'the', 'first', 'option', 'would', 'be', 'to', 'have', 'a', 'retreat', 'similar', 'to', 'what', 'we', \"'\", 've', 'done', 'the', 'past', 'several', 'years', '.', 'this', 'year', 'we', 'could', 'go', 'to', 'the', 'heartland', 'country', 'inn', '(', 'www', '.', '.', 'com', ')', 'outside', 'of', 'brenham', '.', 'it', \"'\", 's', 'a', 'nice', 'place', ',', 'where', 'we', \"'\", 'd', 'have', 'a', '13', '-', 'bedroom', 'and', 'a', '5', '-', 'bedroom', 'house', 'side', 'by', 'side', '.', 'it', \"'\", 's', 'in', 'the', 'country', ',', 'real', 'relaxing', ',', 'but', 'also', 'close', 'to', 'brenham', 'and', 'only', 'about', 'one', 'hour', 'and', '15', 'minutes', 'from', 'here', '.', 'we', 'can', 'golf', ',', 'shop', 'in', 'the', 'antique', 'and', 'craft', 'stores', 'in', 'brenham', ',', 'eat', 'dinner', 'together', 'at', 'the', 'ranch', ',', 'and', 'spend', 'time', 'with', 'each', 'other', '.', 'we', \"'\", 'd', 'meet', 'on', 'saturday', ',', 'and', 'then', 'return', 'on', 'sunday', 'morning', ',', 'just', 'like', 'what', 'we', \"'\", 've', 'done', 'in', 'the', 'past', '.', 'the', 'second', 'option', 'would', 'be', 'to', 'stay', 'here', 'in', 'houston', ',', 'have', 'dinner', 'together', 'at', 'a', 'nice', 'restaurant', ',', 'and', 'then', 'have', 'dessert', 'and', 'a', 'time', 'for', 'visiting', 'and', 'recharging', 'at', 'one', 'of', 'our', 'homes', 'on', 'that', 'saturday', 'evening', '.', 'this', 'might', 'be', 'easier', ',', 'but', 'the', 'trade', 'off', 'would', 'be', 'that', 'we', 'wouldn', \"'\", 't', 'have', 'as', 'much', 'time', 'together', '.', 'i', \"'\", 'll', 'let', 'you', 'decide', '.', 'email', 'me', 'back', 'with', 'what', 'would', 'be', 'your', 'preference', ',', 'and', 'of', 'course', 'if', 'you', \"'\", 're', 'available', 'on', 'that', 'weekend', '.', 'the', 'democratic', 'process', 'will', 'prevail', '-', '-', 'majority', 'vote', 'will', 'rule', '!', 'let', 'me', 'hear', 'from', 'you', 'as', 'soon', 'as', 'possible', ',', 'preferably', 'by', 'the', 'end', 'of', 'the', 'weekend', '.', 'and', 'if', 'the', 'vote', 'doesn', \"'\", 't', 'go', 'your', 'way', ',', 'no', 'complaining', 'allowed', '(', 'like', 'i', 'tend', 'to', 'do', '!', ')', 'have', 'a', 'great', 'weekend', ',', 'great', 'golf', ',', 'great', 'fishing', ',', 'great', 'shopping', ',', 'or', 'whatever', 'makes', 'you', 'happy', '!', 'bobby'], ['Subject', ':', 'photoshop', ',', 'windows', ',', 'office', '.', 'cheap', '.', 'main', 'trending', 'abasements', 'darer', 'prudently', 'fortuitous', 'undergone', 'lighthearted', 'charm', 'orinoco', 'taster', 'railroad', 'affluent', 'pornographic', 'cuvier', 'irvin', 'parkhouse', 'blameworthy', 'chlorophyll', 'robed', 'diagrammatic', 'fogarty', 'clears', 'bayda', 'inconveniencing', 'managing', 'represented', 'smartness', 'hashish', 'academies', 'shareholders', 'unload', 'badness', 'danielson', 'pure', 'caffein', 'spaniard', 'chargeable', 'levin'], ['Subject', ':', 're', ':', 'indian', 'springs', 'this', 'deal', 'is', 'to', 'book', 'the', 'teco', 'pvr', 'revenue', '.', 'it', 'is', 'my', 'understanding', 'that', 'teco', 'just', 'sends', 'us', 'a', 'check', ',', 'i', 'haven', \"'\", 't', 'received', 'an', 'answer', 'as', 'to', 'whether', 'there', 'is', 'a', 'predermined', 'price', 'associated', 'with', 'this', 'deal', 'or', 'if', 'teco', 'just', 'lets', 'us', 'know', 'what', 'we', 'are', 'giving', '.', 'i', 'can', 'continue', 'to', 'chase', 'this', 'deal', 'down', 'if', 'you', 'need', '.']] \n",

STEP 2: REMOVING PUNCTUATIONS AND CASE CONVERSION.

The next step of our model is gathering tokenized words and removing the punctuations and converting the tokens to the lowercase for the learning ability of our model and printing the same

"After Removing punctuation and converting to lowercase : [['subject', 'enron', 'methanol', 'meter', 'this', 'is', 'a', 'follow', 'up', 'to', 'the', 'note', 'i', 'gave', 'you', 'on', 'monday', 'preliminary', 'flow', 'data', 'provided', 'by', 'daren', 'please', 'override', 'pop', 's', 'daily', 'volume', 'presently', 'zero', 'to', 'reflect', 'daily', 'activity', 'you', 'can', 'obtain', 'from', 'gas', 'control', 'this', 'change', 'is', 'needed', 'asap', 'for', 'economics', 'purposes'], ['subject', 'hpl', 'nom', 'for', 'january', 'see', 'attached', 'file', 'hplnol', 'xls', 'hplnol', 'xls'], ['subject', 'neon', 'retreat', 'ho', 'ho', 'ho', 'we', 're', 'around', 'to', 'that', 'most', 'wonderful', 'time', 'of', 'the', 'year', 'neon', 'leaders', 'retreat', 'time', 'i', 'know', 'that', 'this', 'time', 'of', 'year', 'is', 'extremely', 'hectic', 'and', 'that', 'it', 's', 'tough', 'to', 'think', 'about', 'anything', 'past', 'the', 'holidays', 'but', 'life', 'does', 'go', 'on', 'past', 'the', 'week', 'of', 'december', 'through', 'january', 'and', 'that', 's', 'what', 'i', 'd', 'like', 'you', 'to', 'think', 'about', 'for', 'a', 'minute', 'on', 'the', 'calender', 'that', 'i', 'handed', 'out', 'at', 'the', 'beginning', 'of', 'the', 'fall', 'semester', 'the', 'retreat', 'was', 'scheduled', 'for', 'the', 'weekend', 'of', 'january', 'but', 'because', 'of', 'a', 'youth', 'ministers', 'conference', 'that', 'brad', 'and', 'dustin', 'are', 'connected', 'with', 'that', 'week', 'we', 're', 'going', 'to', 'change', 'the', 'date', 'to', 'the', 'following', 'weekend', 'january', 'now', 'comes', 'the', 'part', 'you', 'need', 'to', 'think', 'about', 'i', 'think', 'we', 'all', 'agree', 'that', 'it', 's', 'important', 'for', 'us', 'to', 'get', 'together', 'and', 'have', 'some', 'time', 'to', 'recharge', 'our', 'batteries', 'before', 'we', 'get', 'to', 'far', 'into', 'the', 'spring', 'semester', 'but', 'it', 'can', 'be', 'a', 'lot', 'of', 'trouble', 'and', 'difficult', 'for', 'us', 'to', 'get', 'away', 'without', 'kids', 'etc', 'so', 'brad', 'came', 'up', 'with', 'a', 'potential', 'alternative', 'for', 'how', 'we', 'can', 'get', 'together', 'on', 'that', 'weekend', 'and', 'then', 'you', 'can', 'let', 'me', 'know', 'which', 'you', 'prefer', 'the', 'first', 'option', 'would', 'be', 'to', 'have', 'a', 'retreat', 'similar', 'to', 'what', 'we', 've', 'done', 'the', 'past', 'several', 'years', 'this', 'year', 'we', 'could', 'go', 'to', 'the', 'heartland', 'country', 'inn', 'www', 'com', 'outside', 'of', 'brenham', 'it', 's', 'a', 'nice', 'place', 'where', 'we', 'd', 'have', 'a', 'bedroom', 'and', 'a', 'bedroom', 'house', 'side', 'by', 'side', 'it', 's', 'in', 'the', 'country', 'real', 'relaxing', 'but', 'also', 'close', 'to', 'brenham', 'and', 'only', 'about', 'one', 'hour', 'and', 'minutes', 'from', 'here', 'we', 'can', 'golf', 'shop', 'in', 'the', 'antique', 'and', 'craft', 'stores', 'in', 'brenham', 'eat', 'dinner', 'together', 'at', 'the', 'ranch', 'and', 'spend', 'time', 'with', 'each', 'other', 'we', 'd', 'meet', 'on', 'saturday', 'and', 'then', 'return', 'on', 'sunday', 'morning', 'just', 'like', 'what', 'we', 've', 'done', 'in', 'the', 'past', 'the', 'second', 'option', 'would', 'be', 'to', 'stay', 'here', 'in', 'houston', 'have', 'dinner', 'together', 'at', 'a', 'nice', 'restaurant', 'and', 'then', 'have', 'dessert', 'and', 'a', 'time', 'for', 'visiting', 'and', 'recharging', 'at', 'one', 'of', 'our', 'homes', 'on', 'that', 'saturday', 'evening', 'this', 'might', 'be', 'easier', 'but', 'the', 'trade', 'off', 'would', 'be', 'that', 'we', 'wouldn', 't', 'have', 'as', 'much', 'time', 'together', 'i', 'll', 'let', 'you', 'decide', 'email', 'me', 'back', 'with', 'what', 'would', 'be', 'your', 'preference', 'and', 'of', 'course', 'if', 'you', 're', 'available', 'on', 'that', 'weekend', 'the', 'democratic', 'process', 'will', 'prevail', 'majority', 'vote', 'will', 'rule', 'let', 'me', 'hear', 'from', 'you', 'as', 'soon', 'as', 'possible', 'preferably', 'by', 'the', 'end', 'of', 'the', 'weekend', 'and', 'if', 'the', 'vote', 'doesn', 't', 'go', 'your', 'way', 'no', 'complaining', 'allowed', 'like', 'i', 'tend', 'to', 'do', 'have',

'a', 'great', 'weekend', 'great', 'golf', 'great', 'fishing', 'great', 'shopping', 'or', 'whatever', 'makes', 'you', 'happy', 'bobby'], ['subject', 'photoshop', 'windows', 'office', 'cheap', 'main', 'trending', 'abasements', 'darer', 'prudently', 'fortuitous', 'undergone', 'lighthearted', 'charm', 'orinoco', 'taster', 'railroad', 'affluent', 'pornographic', 'cuvier', 'irvin', 'parkhouse', 'blameworthy', 'chlorophyll', 'robed', 'diagrammatic', 'fogarty', 'clears', 'bayda', 'inconveniencing', 'managing', 'represented', 'smartness', 'hashish', 'academies', 'shareholders', 'unload', 'badness', 'danielson', 'pure', 'caffein', 'spaniard', 'chargeable', 'levin'], ['subject', 're', 'indian', 'springs', 'this', 'deal', 'is', 'to', 'book', 'the', 'teco', 'pvr', 'revenue', 'it', 'is', 'my', 'understanding', 'that', 'teco', 'just', 'sends', 'us', 'a', 'check', 'i', 'haven', 't', 'received', 'an', 'answer', 'as', 'to', 'whether', 'there', 'is', 'a', 'predermined', 'price', 'associated', 'with', 'this', 'deal', 'or', 'if', 'teco', 'just', 'lets', 'us', 'know', 'what', 'we', 'are', 'giving', 'i', 'can', 'continue', 'to', 'chase', 'this', 'deal', 'down', 'if', 'you', 'need']] \n",

STEP 3:STOPWORDS REMOVAL:

After removing the punctuations and case conversion stop words has been removed from the same and printing the same.

"After Removing punctuation and converting to lowercase : [['subject', 'enron', 'methanol', 'meter', 'this', 'is', 'a', 'follow', 'up', 'to', 'the', 'note', 'i', 'gave', 'you', 'on', 'monday', 'preliminary', 'flow', 'data', 'provided', 'by', 'daren', 'please', 'override', 'pop', 's', 'daily', 'volume', 'presently', 'zero', 'to', 'reflect', 'daily', 'activity', 'you', 'can', 'obtain', 'from', 'gas', 'control', 'this', 'change', 'is', 'needed', 'asap', 'for', 'economics', 'purposes'], ['subject', 'hpl', 'nom', 'for', 'january', 'see', 'attached', 'file', 'hplnol', 'xls', 'hplnol', 'xls'], ['subject', 'neon', 'retreat', 'ho', 'ho', 'ho', 'we', 're', 'around', 'to', 'that', 'most', 'wonderful', 'time', 'of', 'the', 'year', 'neon', 'leaders', 'retreat', 'time', 'i', 'know', 'that', 'this', 'time', 'of', 'year', 'is', 'extremely', 'hectic', 'and', 'that', 'it', 's', 'tough', 'to', 'think', 'about', 'anything', 'past', 'the', 'holidays', 'but', 'life', 'does', 'go', 'on', 'past', 'the', 'week', 'of', 'december', 'through', 'january', 'and', 'that', 's', 'what', 'i', 'd', 'like', 'you', 'to', 'think', 'about', 'for', 'a', 'minute', 'on', 'the', 'calender', 'that', 'i', 'handed', 'out', 'at', 'the', 'beginning', 'of', 'the', 'fall', 'semester', 'the', 'retreat', 'was', 'scheduled', 'for', 'the', 'weekend', 'of', 'january', 'but', 'because', 'of', 'a', 'youth', 'ministers', 'conference', 'that', 'brad', 'and', 'dustin', 'are', 'connected', 'with', 'that', 'week', 'we', 're', 'going', 'to', 'change', 'the', 'date', 'to', 'the', 'following', 'weekend', 'january', 'now', 'comes', 'the', 'part', 'you', 'need', 'to', 'think', 'about', 'i', 'think', 'we', 'all', 'agree', 'that', 'it', 's', 'important', 'for', 'us', 'to', 'get', 'together', 'and', 'have', 'some', 'time', 'to', 'recharge', 'our', 'batteries', 'before', 'we', 'get', 'to', 'far', 'into', 'the', 'spring', 'semester', 'but', 'it', 'can', 'be', 'a', 'lot', 'of', 'trouble', 'and', 'difficult', 'for', 'us', 'to', 'get', 'away', 'without', 'kids', 'etc', 'so', 'brad', 'came', 'up', 'with', 'a', 'potential', 'alternative', 'for', 'how', 'we', 'can', 'get', 'together', 'on', 'that', 'weekend', 'and', 'then', 'you', 'can', 'let', 'me', 'know', 'which', 'you', 'prefer', 'the', 'first', 'option', 'would', 'be', 'to', 'have', 'a', 'retreat', 'similar', 'to', 'what', 'we', 've', 'done', 'the', 'past', 'several', 'years', 'this', 'year', 'we', 'could', 'go', 'to', 'the', 'heartland', 'country', 'inn', 'www', 'com', 'outside', 'of', 'brenham', 'it', 's', 'a', 'nice', 'place', 'where', 'we', 'd', 'have', 'a', 'bedroom', 'and', 'a', 'bedroom', 'house', 'side', 'by', 'side', 'it', 's', 'in', 'the', 'country', 'real', 'relaxing', 'but', 'also', 'close', 'to', 'brenham', 'and', 'only', 'about', 'one', 'hour', 'and', 'minutes', 'from', 'here', 'we', 'can', 'golf', 'shop', 'in', 'the', 'antique', 'and', 'craft', 'stores', 'in', 'brenham', 'eat', 'dinner', 'together', 'at', 'the', 'ranch', 'and', 'spend', 'time', 'with', 'each', 'other', 'we', 'd', 'meet', 'on', 'saturday', 'and', 'then', 'return', 'on', 'sunday', 'morning', 'just', 'like', 'what', 'we', 've', 'done', 'in', 'the', 'past', 'the', 'second', 'option', 'would', 'be', 'to', 'stay', 'here', 'in', 'houston', 'have', 'dinner', 'together', 'at', 'a', 'nice', 'restaurant', 'and', 'then', 'have', 'dessert', 'and', 'a', 'time', 'for', 'visiting', 'and', 'recharging', 'at', 'one', 'of', 'our', 'homes', 'on', 'that', 'saturday', 'evening', 'this', 'might', 'be', 'easier', 'but', 'the', 'trade', 'off', 'would', 'be', 'that', 'we', 'wouldn', 't', 'have', 'as', 'much', 'time', 'together', 'i', 'll', 'let', 'you', 'decide', 'email', 'me', 'back', 'with', 'what', 'would', 'be', 'your', 'preference', 'and', 'of', 'course', 'if', 'you', 're', 'available', 'on', 'that', 'weekend', 'the', 'democratic', 'process', 'will', 'prevail', 'majority', 'vote', 'will', 'rule', 'let', 'me', 'hear', 'from', 'you', 'as', 'soon', 'as', 'possible', 'preferably', 'by', 'the', 'end', 'of', 'the', 'weekend',

'and', 'if', 'the', 'vote', 'doesn', 't', 'go', 'your', 'way', 'no', 'complaining', 'allowed', 'like', 'i', 'tend', 'to', 'do', 'have', 'a', 'great', 'weekend', 'great', 'golf', 'great', 'fishing', 'great', 'shopping', 'or', 'whatever', 'makes', 'you', 'happy', 'bobby'], ['subject', 'photoshop', 'windows', 'office', 'cheap', 'main', 'trending', 'abasements', 'darer', 'prudently', 'fortuitous', 'undergone', 'lighthearted', 'charm', 'orinoco', 'taster', 'railroad', 'affluent', 'pornographic', 'cuvier', 'irvin', 'parkhouse', 'blameworthy', 'chlorophyll', 'robed', 'diagrammatic', 'fogarty', 'clears', 'bayda', 'inconveniencing', 'managing', 'represented', 'smartness', 'hashish', 'academies', 'shareholders', 'unload', 'badness', 'danielson', 'pure', 'caffein', 'spaniard', 'chargeable', 'levin'], ['subject', 're', 'indian', 'springs', 'this', 'deal', 'is', 'to', 'book', 'the', 'teco', 'pvr', 'revenue', 'it', 'is', 'my', 'understanding', 'that', 'teco', 'just', 'sends', 'us', 'a', 'check', 'i', 'haven', 't', 'received', 'an', 'answer', 'as', 'to', 'whether', 'there', 'is', 'a', 'predermined', 'price', 'associated', 'with', 'this', 'deal', 'or', 'if', 'teco', 'just', 'lets', 'us', 'know', 'what', 'we', 'are', 'giving', 'i', 'can', 'continue', 'to', 'chase', 'this', 'deal', 'down', 'if', 'you', 'need']] \n",

STEP 4:STEMMING:

After removing the unnecessary tokens from the data it has been convert to its root forms using stemming and printing the same.

"AFTER STEMMING: [['subject', 'enron', 'methanol', 'meter', 'follow', 'note', 'gave', 'monday', 'preliminari', 'flow', 'data', 'provid', 'daren', 'pleas', 'overrid', 'pop', 'daili', 'volum', 'present', 'zero', 'reflect', 'daili', 'activ', 'obtain', 'ga', 'control', 'chang', 'need', 'asap', 'econom', 'purpos'], ['subject', 'hpl', 'nom', 'januari', 'see', 'attach', 'file', 'hplnol', 'xl', 'hplnol', 'xl'], ['subject', 'neon', 'retreat', 'ho', 'ho', 'ho', 'around', 'wonder', 'time', 'year', 'neon', 'leader', 'retreat', 'time', 'know', 'time', 'year', 'extrem', 'hectic', 'tough', 'think', 'anyth', 'past', 'holiday', 'life', 'go', 'past', 'week', 'decemb', 'januari', 'like', 'think', 'minut', 'calend', 'hand', 'begin', 'fall', 'semest', 'retreat', 'schedul', 'weekend', 'januari', 'youth', 'minist', 'confer', 'brad', 'dustin', 'connect', 'week', 'go', 'chang', 'date', 'follow', 'weekend', 'januari', 'come', 'part', 'need', 'think', 'think', 'agre', 'import', 'us', 'get', 'togeth', 'time', 'recharg', 'batteri', 'get', 'far', 'spring', 'semest', 'lot', 'troubl', 'difficult', 'us', 'get', 'away', 'without', 'kid', 'etc', 'brad', 'came', 'potenti', 'altern', 'get', 'togeth', 'weekend', 'let', 'know', 'prefer', 'first', 'option', 'would', 'retreat', 'similar', 'done', 'past', 'sever', 'year', 'year', 'could', 'go', 'heartland', 'countri', 'inn', 'www', 'com', 'outsid', 'brenham', 'nice', 'place', 'bedroom', 'bedroom', 'hous', 'side', 'side', 'countri', 'real', 'relax', 'also', 'close', 'brenham', 'one', 'hour', 'minut', 'golf', 'shop', 'antiqu', 'craft', 'store', 'brenham', 'eat', 'dinner', 'togeth', 'ranch', 'spend', 'time', 'meet', 'saturday', 'return', 'sunday', 'morn', 'like', 'done', 'past', 'second', 'option', 'would', 'stay', 'houston', 'dinner', 'togeth', 'nice', 'restaur', 'dessert', 'time', 'visit', 'recharg', 'one', 'home', 'saturday', 'even', 'might', 'easier', 'trade', 'would', 'much', 'time', 'togeth', '',ecid' let'd, 'email', 'back', 'would', 'prefer', 'cours', 'avail', 'weekend', 'democrat', 'process', 'prevail', 'major', 'vote', 'rule', 'let', 'hear', 'soon', 'possibl', 'prefer', 'end', 'weekend', 'vote', 'go', 'way', 'complain', 'allow', 'like', 'tend', 'great', 'weekend', 'great', 'golf', 'great', 'fish', 'great', 'shop', 'whatev', 'make', 'happi', 'bobbi'], ['subject', 'photoshop', 'window', 'offic', 'cheap', 'main', 'trend', 'abas', 'darer', 'prudent', 'fortuit', 'undergon', 'lightheart', 'charm', 'orinoco', 'taster', 'railroad', 'affluent', 'pornograph', 'cuvier', 'irvin', 'parkhous', 'blameworthi', 'chlorophyl', 'robe', 'diagrammat', 'fogarti', 'clear', 'bayda', 'manag'inconvenienc',', 'repres', 'smart', 'hashish', 'academi', 'sharehold', 'unload', 'bad', 'danielson', 'pure', 'caffein', 'spaniard', 'chargeabl', 'levin'], ['subject', 'indian', 'spring', 'deal', 'book', 'teco', 'pvr', 'revenu', 'understand', 'teco', 'send', 'us', 'check', 'receiv', 'answer', 'whether', 'predermin', 'price', 'associ', 'deal', 'teco', 'let', 'us', 'know', 'give', 'continu', 'chase', 'deal', 'need']]

FINAL STEP:

After doing all the pre-processing steps the final pre-processed output has been printed.

AFTER PREPROCESSING:

0 subject enron methanol meter follow note gave.....

1 subject hpl non januari see attach file.....

2 subject neon retreat ho ho ho around wonder ti.....

3 subject Photoshop window offic cheap main tren.....

4 subject indian spring deal book teco pvr reven.....

PREPROCESSING CODE:

LANGUAGE: PYTHON

PLATFORM: JUPYTER NOTEBOOK

```
import pandas as pd
#from sklearn.model_selection import train_test_split
#from sklearn.feature_extraction.text import CountVectorizer
#from sklearn.ensemble import RandomForestClassifier
#from sklearn.metrics import accuracy_score, classification_report
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

# Load the dataset
data = pd.read_csv("C:/Users/HP/Downloads/spam_ham_dataset.csv")
df = pd.DataFrame(data)

# Preprocess the text data

totaltokens = []
totalpunc =[]
totalstopwords=[]
totalstem=[]

def preprocess_text(text):
```

```
# Tokenization

global totaltokens

tokens = word_tokenize(text)

totaltokens.append(tokens)

#print('AFTER TOKENIZATION:', totaltokens, "\n")



# Removing punctuation and converting to lowercase

global totalpunc

tokens1 = [word.lower() for word in tokens if word.isalpha()]

totalpunc.append(tokens1)

#print('After Removing punctuation and converting to lowercase :', totalpunc, "\n")

# Removing stopwords

global totalstopwords

stop_words = set(stopwords.words("english"))

tokens2 = [word for word in tokens1 if word not in stop_words]

totalstopwords.append(tokens2)

#print('AFTER REMOVING STOPWORDS:', totalstopwords, "\n")

# Stemming

global totalstem

stemmer = PorterStemmer()

tokens3 = [stemmer.stem(word) for word in tokens2]

totalstem.append  (tokens3)

#print('AFTER STEMMING:', totalstem, "\n")

# Join tokens back into a string
```

```
    preprocessed_text = " ".join(tokens3)

    #print('AFTER JOINING:', preprocessed_text)


    return preprocessed_text


# Apply the preprocess_text function to the 'text' column

df['text'] = df['text'].head().apply(preprocess_text)


# Print the first few rows of the preprocessed text

print("AFTER PREPROCESSING:\n",df['text'].head())

print('AFTER TOKENIZATION:', totaltokens, "\n")

print('After Removing punctuation and converting to lowercase :', totalpunc, "\n")

print('AFTER REMOVING STOPWORDS:', totalstopwords, "\n")

print('AFTER STEMMING:', totalstem, "\n")
```

Feature Extraction:

Bag of Words (BoW):

* Tokenization: Split the text into individual words or tokens.

*Stop Words Removal: Remove common words (e.g., "the," "and," "is") that may not be informative.

*Stemming or Lemmatization: Reduce words to their root form (e.g., "running" to "run").

Machine Learning Algorithm:

* Random Forest

*Voting Classifier

Training model

STEP 1: WE HAVE TO IMPORT LIBRARY FOR USING ALGORITHM SUCH AS RANDOMFOREST CLASSIFIER AND VOTING CLASSIFIER

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.ensemble import RandomForestClassifier, VotingClassifier
```

STEP 2:WE HAVE TO SPLIT THE DATASET INTO TRAINING AND TESTING SETS

```
X = data['text']

y = data['label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

STEP 3:WE HAVE TO VECTORIZE THE DATASET USING COUNTERVECTORIZER

```
vectorizer = CountVectorizer()

X_train = vectorizer.fit_transform(X_train)

X_test = vectorizer.transform(X_test)
```

STEP 4:WE HAVE TO TRAIN THE MODEL

```
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

ensemble_classifier = VotingClassifier(estimators=[('rf', rf_classifier)], voting='hard')


ensemble_classifier.fit(X_train, y_train)

print(X_train)
```

FINAL STEP:AFTER TRAINING THE MODEL THE FINAL OUTPUT HAS BEEN PRINTED

```
(0, 28704) 1

(0, 1553) 3

(0, 291) 3

(0, 28918) 6
```

(0, 28349) 1

(0, 6330) 2

(0, 7851) 2

(0, 1991) 2

(0, 9292) 1

(0, 7014) 1

(0, 6430) 1

(0, 29384) 2

(0, 23327) 4

(0, 26823) 1

(0, 24908) 2

(0, 11438) 3

(0, 9296)3

(0, 24570) 1

(0, 25370) 1

(0, 20129) 4

(0, 11228) 1

(0, 21354) 1

(0, 10204) 1

(0, 7888) 1

(0, 29832) 1

: :

(4135, 5173) 2

(4135, 19523) 4

(4135, 7978) 3

(4135, 26958) 3

(4135, 12665) 2

(4135, 14539) 2

(4135, 9705) 4

(4135, 30090) 1

(4135, 11677) 1

(4135, 671) 4

(4135, 17535) 2

(4135, 4946) 1

(4135, 16641) 3

(4135, 23375) 2

(4135, 10176) 1

(4135, 7810) 3

(4135, 10806) 1

(4135, 972) 1

(4135, 31247) 1

(4135, 11118) 2

(4135, 17582) 1

(4135, 25659) 1

(4135, 9067) 1

(4135, 23522) 1

(4135, 33457) 1

TRAINING CODE :

LANGUAGE :PYTHON

PLATFORM:JUPYTER NOTEBOOK

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split
```

```
from sklearn.feature_extraction.text import CountVectorizer

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

#from sklearn.metrics import accuracy_score, classification_report

#from nltk.corpus import stopwords

#from nltk.stem import PorterStemmer

#from nltk.tokenize import word_tokenize


# Load the dataset

data = pd.read_csv("spam_ham_dataset.csv")


# Preprocess the text data

def preprocess_text(text):

# Tokenization

tokens = word_tokenize(text)


# Removing punctuation and converting to lowercase

tokens = [word.lower() for word in tokens if word.isalpha()]


# Removing stopwords

stop_words = set(stopwords.words("english"))

tokens = [word for word in tokens if word not in stop_words]


# Stemming

stemmer = PorterStemmer()

tokens = [stemmer.stem(word) for word in tokens]


# Join tokens back into a string

preprocessed_text = " ".join(tokens)
```

```
return preprocessed_text


data['text'] = data['text'].apply(preprocess_text)


# Split the dataset into training and testing sets

X = data['text']

y = data['label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Vectorize the text data using CountVectorizer

vectorizer = CountVectorizer()

X_train = vectorizer.fit_transform(X_train)

X_test = vectorizer.transform(X_test)


# Train an ensemble model

rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

ensemble_classifier = VotingClassifier(estimators=[('rf', rf_classifier)], voting='hard')


ensemble_classifier.fit(X_train, y_train)

print(X_train)
```

Evaluation metrices:

*Accuracy

*Precision

*Recall

*Mean absolute error

*Root mean squared error

Innovative Techniques:

1. Diversity in Base Models:

Ensure that the base models in your ensemble are diverse in their algorithms and feature representations. This can include using different machine learning algorithms (e.g., Naive Bayes, Random Forest, Support Vector Machines, Neural Networks) and different types of features (e.g., text-based features, header information, sender reputation, time-based features). Diversity helps in capturing different aspects of spam patterns.

2. Meta-Ensemble Models:

Create a meta-ensemble by training an additional model (meta-learner) on the predictions of the base models. This meta-learner can learn to combine the predictions effectively and might provide better performance. Common meta-learners include stacking, bagging, and boosting techniques.

3. Dynamic Weighting:

Assign different weights to the base models based on their performance on a validation set. Models that perform better on specific subsets of data can be given higher weights, allowing the ensemble to adapt to different types of spam.