Programming Exercise 2 Technical Document

Winson Ma

Description: This code analyzes an email message that is proved by the user and determines if the message is spam. It uses a list of common keywords and calculates a “spam score” and then rates the message’s spam likelihood based on that score.

**Function**: Calculate\_spam\_score(message, spam\_keywords)- Calculates the spam score of an email by counting spam keywords.

* Parameters:
* Message (string): email message to be analyzed.
* Spam\_keywords(list of strings): List of keywords commonly found in spams.
* Variables:
* Score (integer): initializes to 0, stores the calculated spam score.
* Triggered (list of tuples): Stores the keywords that triggered the spam score.
* Matches (list of strings): Stores results of the expression search for each word.
* Logical steps:

1. Initialize score to 0 and triggered to an empty list.
2. Iterates through each keyword.
3. Uses re.findall() with word boundaries and case-insensitive matching(re.IGNORECASE) to find all occurrences of the keyword in the spam.
4. Adds the number of matches or found to the score.
5. If there are any matches for keywords, it appends a tuple (keyword, len(matches)) to the list.
6. After iterating, the function returns the score and triggered list.

**Returns:**

* Score (int): Calculated spam.
* Triggered (list of tuples): each tuple contains a triggered keyword and its count.

**Functions:** Assess\_spam\_likelihood(score)- rates the likelihood of an email being spam based on the calculated spam score.

* Parameters:
* Score (int): Spam is calculated by calculate\_spam\_score.
* Logical Steps:

1. Uses if-elif-else statements to determine the spam likelihood based on score.

Returns:

* (Str): representing spam likelihood.

Function: Main()- controls the program’s execution.

* Variables:
* Spam\_keywords (list of str): List of predefined keywords.
* Message (str): Stores message input by user.
* Score (int): Stores spam score.
* Triggered (list of tuples): stores triggered keywords return by calculate\_spam\_score.
* Likelihood (str): Stores the spam likelihood returned by assess\_spam\_likelihood.
* Logical steps:

1. Defines the spam\_keywords list.
2. Ask user to enter an email and stores it in the message variable.
3. Calls calculate\_spam\_score with the message and spam\_keywords to get the spam score and triggered keywords.
4. Calls assess\_spame\_likelihood with the spam score to determine the spam.
5. Prints the spam score.
6. If triggered list is not empty, it prints keywords and their counts.
7. If no keywords indicated, it prints no spam keywords were found.

Returns:

* None.

GitHub: <https://github.com/FamiliarotherW/COP2373-Assignment-02>

A screenshot of a computer screen

AI-generated content may be incorrect.