**Spam Detection using Machine Learning**

**By Thej Venkat P**

This is a Python program that uses a Machine Learning model to classify text messages as spam or ham (not spam). The program also can detect Harmful or Dangerous URLs present in the Message and Shows if they are Dangerous.

Github link: [Thej-Venkat-P/Spam-Message-Checker: Checks for spam Messages and Dangerous URLs (github.com)](https://github.com/Thej-Venkat-P/Spam-Message-Checker)

**Requirements**

The program requires the following files and libraries to be in the same directory or imported:

* Python Programming Language with an IDE
* Dataset for Training the Machine Learning model
* Python File with functions to check for Safety of URL
* Pandas module
* Sklearn module
* Re module

**Setup Instructions**

To run this program, you will need the following:

* Python 3.8 or higher ([Download Python | Python.org](https://www.python.org/downloads/))
* The DataSet file “spam.csv”, which contains the data for training and testing the Machine learning model.
* The Python file “urlSafety.py”, which contains the function for checking the safety of a URL.
* The following Python libraries: pandas, sklearn, and re. You can install them using the command **“pip install pandas sklearn“** in the Command Terminal.

**Brief Explanation of the Code**

The Program code can be found at: [Spam-Message-Checker/spamChecker.py](https://github.com/Thej-Venkat-P/Spam-Message-Checker/blob/main/spamChecker.py)

The Video Showing the Working of the Code can be Downloaded at: [spamChecker.mp4](https://github.com/Thej-Venkat-P/Spam-Message-Checker/blob/main/spamChecker.mp4)

Bried Explanation of the Code is given as follows:

1. Import necessary libraries:

- `pandas` for data manipulation and analysis.

- `train\_test\_split` from `sklearn.model\_selection` for splitting the dataset into training and testing sets.

- `CountVectorizer` from `sklearn.feature\_extraction.text` for converting text data into a matrix of token counts.

- `svm` from `sklearn` for Support Vector Machine model.

- `checkLink` from `urlSafety` for checking the safety of URLs.

- `re` for regular expression operations.

2. Read the spam dataset from a CSV file and handle missing values.

3. Preprocess the data:

- Extract the "Message" and "Category" columns from the dataset.

- Use `CountVectorizer` to convert the text data into a matrix of token counts.

4. Train the Support Vector Machine (SVM) model using the features and categories.

5. Function `find\_urls` to extract URLs from a given message using regular expressions.

6. Function `checkSpam` to check if a given message is spam or not:

- Convert the message to lowercase.

- Use the trained model to predict if the message is spam or not.

- If the message contains URLs, check the safety of each URL using the `checkLink` function.

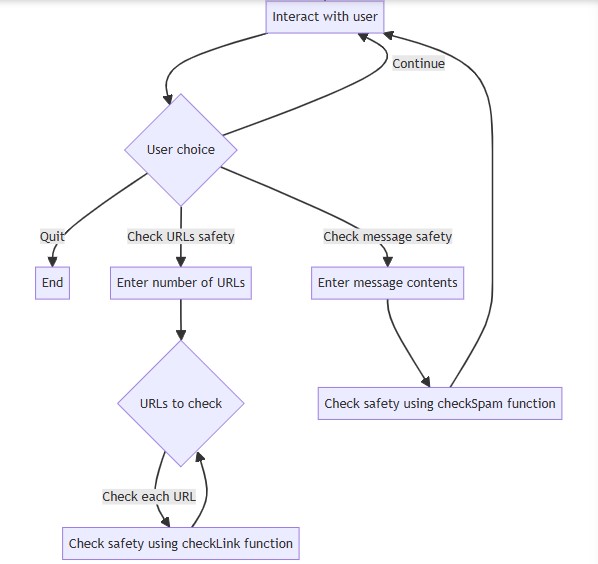
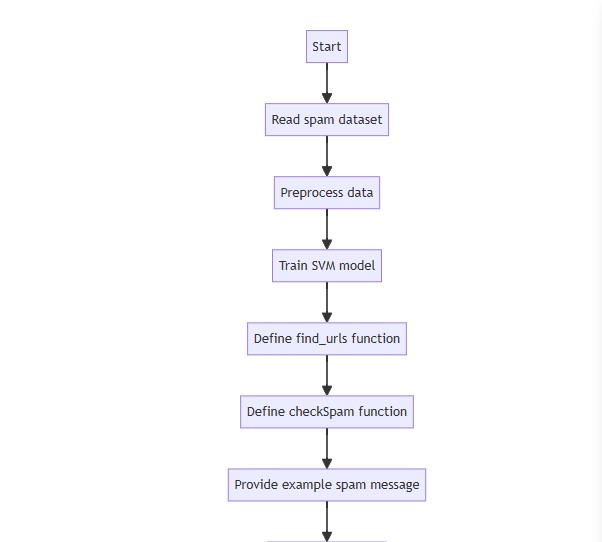
7. Provide an example spam message and check its safety using the `checkSpam` function.

8. Implement a loop to interact with the user:

- Allow the user to choose between quitting, checking the safety of URLs, or checking the safety of a message.

- If the user chooses to check the safety of URLs, prompt for the number of URLs and then check each URL using the `checkLink` function.

- If the user chooses to check the safety of a message, prompt for the message contents and then check its safety using the `checkSpam` function.

**FLOWCHART**