TITLE: BUILDING SMARTER AI POWERED SPAM CLASSIFIER

I. Introduction

The objective of this project is to develop a cutting-edge smart AI-powered spam classifier, a critical component in email security. Our innovative approach seeks to tackle the ever-evolving challenges posed by spam, phishing, and other malicious email threats. The primary goal is to enhance email security for users while optimizing their email management experience.

II. Project Overview

In this pre-final year project, we will explore a range of innovative ideas to bolster the capabilities of our AI-powered spam classifier. The project's focus will be on implementing advanced techniques and features that transcend conventional spam classification methods. Below, we outline some of the key innovation ideas

III. Innovative Ideas for the Project

1. Multimodal Analysis: Incorporate advanced text, image, and attachment analysis to detect spam in various formats.

2. Multilingual Support: Ensure the spam classifier can effectively detect spam in multiple languages, catering to a diverse user base.

3. Personalized Classification: Develop a system that adapts to individual user behavior, customizing spam classification for each user.

4. Blockchain for Sender Verification: Investigate the use of blockchain technology to validate email senders and prevent email spoofing.

5. Behavioral Biometrics: Incorporate behavioral biometrics to identify deviations in user behavior as a means of detecting spam.

6. Zero-Day Threat Detection: Utilize advanced machine learning techniques to identify emerging spam tactics and zero-day threats.

7. Natural Language Understanding: Improve the classifier's ability to understand the context and intent of emails to reduce false positives and negatives.

8. AI-Powered Response Suggestions: Offer AI-generated response suggestions for reported spam, empowering users to take informed actions.

9. Email Fingerprinting: Use email fingerprinting to identify and block known spammers or repeat offenders more effectively.

10. Sentiment Analysis: Analyze email sentiments to detect harmful or manipulative content often present in spam.

IV. Project Scope and Deliverables

This project will encompass the development and integration of the aforementioned innovation ideas into a cohesive smart AI-powered spam classifier. Our goal is to create a functional prototype capable of effectively identifying and managing spam emails while enhancing the overall email security experience for users.

V. Project Benefits

The implementation of this project offers a myriad of benefits, including:

• Improved email security for users and organizations.

• Time and productivity savings through automated spam detection.

• User empowerment, knowledge, and trust in email security.

• Reduced risk of falling victim to email-based scams.

• Competitive advantage for organizations prioritizing user-centric email

Security

VI. Conclusion

Our project aims to be at the forefront of email security innovation, making email communication safer and more efficient for all users. The innovative ideas presented in this proposal are foundational to achieving this goal and will be meticulously developed and integrated into our smart AI-powered spam classifier.

We look forward to the opportunity to undertake this project, contribute to the field of email security, and deliver a robust and user-centric solution.

annexure

Program

-\*- coding: utf-8 -\*-

# coding: utf-8

#Naive Bayes

import os

import io

import numpy

from pandas import DataFrame

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

#Function to read files (emails) from the local directory

def readFiles(path):

for root, dirnames, filenames in os.walk(path):

for filename in filenames:

path = os.path.join(root, filename)

inBody = False

lines = []

f = io.open(path, 'r', encoding='latin1')

for line in f:

if inBody:

lines.append(line)

elif line == '\n':

inBody = True

f.close()

message = '\n'.join(lines)

yield path, message

def dataFrameFromDirectory(path, classification):

rows = []

index = []

for filename, message in readFiles(path):

rows.append({'message': message, 'class': classification})

index.append(filename)

return DataFrame(rows, index=index)

#An empty dataframe with 'message' and 'class' headers

data = DataFrame({'message': [], 'class': []})

#Including the email details with the spam/ham classification in the dataframe

data = data.append(dataFrameFromDirectory('C:/Users/surya/Desktop/DecemberBreak/Data Science with Python & R/DataScience/DataScience-Python3/emails/spam', 'spam'))

data = data.append(dataFrameFromDirectory('C:/Users/surya/Desktop/DecemberBreak/Data Science with Python & R/DataScience/DataScience-Python3/emails/ham', 'ham'))

data = data.append(dataFrameFromDirectory('C:/Users/surya/Desktop/DecemberBreak/emails/spam', 'spam'))

data = data.append(dataFrameFromDirectory('C:/Users/surya/Desktop/DecemberBreak/emails/ham', 'ham'))

#Head and the Tail of 'data'

data.head()

print(data.tail())

vectoriser = CountVectorizer()

count = vectoriser.fit\_transform(data['message'].values)

print(count)

target = data['class'].values

print(target)

classifier = MultinomialNB()

classifier.fit(count, target)

print(classifier)

exampleInput = ["Hey. This is John Cena. You can't see me", "Free Viagra boys!!", "Please reply to get this offer"]

excount = vectoriser.transform(exampleInput)

print(excount)

prediction = classifier.predict(excount)

print(prediction)

output:

(0, 20104) 1 [0->1st sentence; 20104->word id; 1-> no. of times that the word occurs in the sentence]

(0, 15629) 1

(0, 30882) 1

(0, 50553) 1

(0, 36099) 1

(0, 44217) 1

(0, 58467) 1

(0, 51216) 1

(0, 10966) 1

(0, 47038) 1

(0, 46816) 1

(0, 54656) 1

(0, 43219) 2

(0, 16635) 1

(0, 38953) 1

(0, 14434) 1

(0, 16777) 1

(0, 36134) 1

(0, 35030) 1

(0, 46819) 1

(0, 12870) 1

(0, 58727) 1

(0, 22787) 1

(0, 22197) 2