Misr University for Science & Technology
College of Information Technology

جامعة مصر للعلوم والتكنولوجيا
كلية تكنولوجيا المعلومات

CS 481 Pattern Recognition

AI spam email filtering

Team members:

sn	Name	ID	Sec
1	Ahmed Khaled Mohamed Ismael Kira	89671	B
2			

Submitted to:

Prof. Dr. Mostafa Gadal-haqq

T.A. Aya Mahmoud

(Fall 2022-2023)

List of Content

Table of Contents

1.1 Overview

Spam emails are a pervasive problem that can have significant negative impacts on individuals and organizations. They are unwanted and unsolicited emails that are often sent in large quantities and contain fraudulent or misleading content. Spam emails can clog up your inbox, waste your time, and even pose a security threat if they contain malicious links or attachments.

The problem of spam emails is particularly acute for businesses and organizations, which can receive large volumes of spam emails and may have limited resources for manually sorting through and filtering them. In addition, spam emails can disrupt productivity and distract employees from more important tasks. Moreover, spam emails can expose businesses to security risks if they contain malicious links or attachments that can compromise the organization's data or systems.

Using artificial intelligence (AI) to filter spam emails can help to reduce the amount of spam that you receive and improve your overall email experience. AI-based spam filters can leverage machine learning algorithms and natural language processing techniques to identify and classify spam emails with high accuracy. This can significantly reduce the amount of spam that reaches your inbox and free up your time to focus on more important tasks. In addition, AI-based spam filters can be trained on new data and fine-tuned over time to improve their accuracy and adapt to new spam tactics.

In this paper, we will explore the use of AI to filter spam emails and discuss the benefits, challenges, and considerations for implementing an AI-based spam filter.

1.2 Problem Statement (Definition)

The problem of spam emails is a significant one, as it can be both annoying and potentially dangerous. Spam emails can take up valuable storage space, consume bandwidth, and distract you from more important tasks. In addition, spam emails can contain malicious links or attachments that can compromise your security if you click on them.

1.3 The Proposed solution

One solution to the problem of spam emails is to use AI to filter them out. There are several ways that AI can be used to detect and filter spam emails, including machine learning algorithms and natural language processing techniques.

1.4 System Tools

a. Software Requirements

SW tools used to implement the solution:

- Python programming language, for implementing the AI algorithm.
- A machine learning library, such as scikit-learn or TensorFlow, for training and evaluating the AI model.
- An email client, such as Microsoft Outlook or Gmail, for integrating the AI filter into your email workflow.

b. Hardware Requirements

Hardware specification for the computer used to run the implementation:

A computer with sufficient processing power and memory to run the AI algorithm and email client with an internet connection for accessing email and training data

1.5 Results and Discussions

Discuss your results and give screenshots for the output of the implementation.

```
from sklearn.metrics import accuracy_score
accuracy_score (y_test,y_pred )
```

```
0.93
```

```
# Make a prediction on the new email (Not Spam example)
new_email = """Hi,

I hope this finds you well. I just wanted to remind you about our meeting to

Best regards,
sara"""
prediction = gnb.predict(vectorizer.transform([new_email]).toarray())

# Print the prediction
print(prediction[0])
print("Spam" if prediction[0] else "Not Spam")
```

```
0
Not Spam
```

```
# Make a prediction on the new email (Spam example)
new_email = """Dear Friend,

I hope you are doing well. I wanted to let you know about a great opportunit

Don't miss out on this chance to change your life. Click the link now!

Sincerely,
Mark"""
prediction = gnb.predict(vectorizer.transform([new_email]).toarray())

# Print the prediction
print(prediction[0])
print("Spam" if prediction[0] else "Not Spam")
```

```
1
Spam
```

1.6 Conclusions

This is a program that uses machine learning to classify emails as spam or not spam (also known as ham). It does this by using a Naive Bayes classifier and training it on a dataset of emails that are labeled as either spam or not. The program first reads in the data from a file called 'spam_or_not_spam.csv' using the Pandas library. It then replaces any missing values (indicated by 'NaN') with an empty string. Next, it uses the CountVectorizer class from the scikit-learn library to convert the emails into a matrix of token counts, where each row represents an email and each column represents a unique word in the vocabulary. The matrix is then split into training and test sets using the train_test_split function from scikit-learn. The program then trains a Gaussian Naive Bayes classifier on the training set using the fit method and makes predictions on the test set using the predict method. Finally, it uses the accuracy_score function from scikit-learn to evaluate the model by comparing the predicted labels to the true labels in the test set.

1.7 Appendix

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
# Read in the data and replace missing values with an empty string
df=pd.read_csv('./spam_or_not_spam.csv')
df=df.replace(np.nan,'')
# df.head()
# Convert emails to a matrix of token counts
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(df['email'])
X=X.toarray()
# Train the classifier
X_train, X_test, y_train, y_test = train_test_split(X, df['label'], test_size=0.2)
gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score (y_test,y_pred )
```

```
# Make a prediction on the new email (Not Spam example)
new_email = """Hi,

I hope this finds you well. I just wanted to remind you about our meeting tomorrow at 10 am in the conference room. Please bring any materials that you have prepared so far.

Best regards,

sara"""
prediction = gnb.predict(vectorizer.transform([new_email]).toarray())
# Print the prediction
print(prediction[0])
print("Spam" if prediction[0] else "Not Spam")
# Make a prediction on the new email (Spam example)
new_email = """Dear Friend,

I hope you are doing well. I wanted to let you know about a great opportunity to make money online. All you have to do is click this link and follow the simple instructions. You can earn thousands of dollars in just a few short weeks!

Don't miss out on this chance to change your life. Click the link now!

Sincerely,

Mark"""
prediction = gnb.predict(vectorizer.transform([new_email]).toarray())
# Print the prediction
print(prediction[0])
print("Spam" if prediction[0] else "Not Spam")
```