



Tech Saksham

Capstone Project Report

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FUNDAMENTALS

EMAIL SPAM DETECTION

KINGSTON ENGINEERING COLLEGE

NM ID	NAME
au511321105003	SRIRAM B

Trainer Name

Master Trainer

RAMAR BOSE

ABSTRACT

Nowadays, a big part of people rely on available email or messages sent by the stranger. The possibility that anybody can leave an email or a message provides a golden opportunity for spammers to write spam message about our different interests .Spam fills inbox with number of ridiculous emails . Degrades our internet speed to a great extent .Steals useful information like our details on our contact list. Identifying these spammers and also the spam content can be a hot topic of research and laborious tasks. Email spam is an operation to send messages in bulk by mail .Since the expense of the spam is borne mostly by the recipient ,it is effectively postage due advertising. Spam email is a kind of commercial advertising which is economically viable because email could be a very cost effective medium for sender .With this proposed model the specified message can be stated as spam or not using Bayes' theorem and Naive Bayes' Classifier and Also IP addresses of the sender are often detected .

INDEX

Sr. No.	Table of Contents	Page No.
1	Chapter 1: Introduction	1
2	Chapter 2: Services and Tools Required	6
3	Chapter 3: Project Architecture	7
4	Chapter 4: Project Outcome	8
5	Conclusion	33
6	Future Scope	34
7	References	35
8	Code	36

CHAPTER 1

INTRODUCTION

Today, Spam has become a major problem in communication over internet. It has been accounted that around 55% of all emails are reported as spam and the number has been growing steadily. Spam which is also known as unsolicited bulk email has led to the increasing use of email as email provides the perfect ways to send the unwanted advertisement or junk newsgroup posting at no cost for the sender. This chances has been extensively exploited by irresponsible organizations and resulting to clutter the mail boxes of millions of people all around the world. Spam has been a major concern given the offensive content of messages, spam is a waste of time. End user is at risk of deleting legitimate mail by mistake. Moreover, spam also impacted the economical which led some countries to adopt legislation. Text classification is used to determine the path of incoming mail/message either into inbox or straight to spam folder. It is the process of assigning categories to text according to its content. It is used to organized, structures and categorize text. It can be done either manually or automatically.

Machine learning automatically classifies the text in a much faster way than manual technique. Machine learning uses pre-labelled text to learn the different associations between pieces of text and it output. It used feature extraction to transform each text to numerical representation in form of vector which represents the frequency of word in predefined dictionary. Text classification is important to structure the unstructured and messy nature of text such as documents and spam messages in a cost-effective way. Machine learning can make more accurate precisions in real-time and help to improve the manual slow process to much better and faster analysing big data. It is important especially to a company to analyse text data, help inform business decisions and even automate business processes. In this project, machine learning techniques are used to detect the spam message of a mail. Machine learning is where computers can learn to do something 10 without the need to explicitly program them for the task. It uses data and produce a program to perform a task such as classification. Compared to knowledge engineering, machine learning techniques require messages that have been successfully pre-classified. The pre-classified messages make the training dataset which will be used to fit the learning algorithm to the model in machine learning studio. A combination of algorithms are used to learn the classification rules from messages. These algorithms are used for classification of objects of different classes. These algorithms are provided with pre labelled data and an unknown text. After learning from the prelabelled data each of these algorithms predict which class the unknown text may belong to and the category predicted by majority is considered as final.

1.1 Problem Statement

Unwanted e-mails irritating internet connection

Critical e-mail message are missed and delayed

Millions of compromised computers

It occupies more space in the cloud

Identity theft

Spam can crash mail servers and fill up hard drives

1.2 Proposed Solution

In this system, to solve the problem of spam, the spam classification system is created to identify spam and nonspam. Since spammers may send spam messages many times, it is difficult to identify it every time manually. So we will be using some of the strategies in our proposed system to detect the spam.

The proposed solution not only identifies the spam word but also identifies the IP address of the system through which the spam message is sent so that next time when the spam message is sent from the same system our proposed system directly identifies it as blacklisted based on the IP address.

In the proposed model, the web application is done using dot net and spam detection is done using machine learning. The web application consists of following modules:

1.3 Feature

Email spam detection relies on a variety of features extracted from email data to distinguish between spam and legitimate messages. These features serve as input variables for machine learning algorithms and statistical models used in spam detection systems. Here are some common features used in email spam detection:

1. **Sender Information:** Characteristics of the email sender, including the sender's email address, domain reputation, sender's IP address, and authentication status (e.g., SPF, DKIM, DMARC). Anomalies or inconsistencies in sender information can indicate potential spam.
2. **Content Analysis:** Analysis of the textual content of the email, including subject line, body text, and embedded links. Features extracted from content analysis may include:
 - Presence of spam-related keywords or phrases (e.g., "free," "discount," "limited time offer").
 - Frequency of certain words or phrases.
 - Use of HTML or rich text formatting.
 - Presence of misspellings, unusual characters, or obfuscation techniques.

3. **Metadata Analysis:** Examination of metadata associated with the email, such as timestamp, message ID, and header information. Metadata features may include:
 - Time of day the email was sent.
 - Geolocation of the sender's IP address.
 - Number of recipients.
 - Email client or software used to send the email.
4. **Structural Analysis:** Analysis of the structural characteristics of the email, including:
 - Number of recipients (to, cc, bcc).
 - Presence of attachments or embedded media files.
 - MIME type of attachments.
 - HTML code analysis for suspicious elements (e.g., hidden text, invisible links).
5. **URL Analysis:** Examination of URLs contained within the email, including:
 - URL length and format.
 - Domain reputation of linked websites.
 - Presence of URL redirects or URL shortening services.
 - Blacklisted or suspicious domains.
6. **Header Analysis:** Inspection of email headers for anomalies or signs of spoofing, including:
 - Consistency between the "From" header and the sender's domain.
 - Presence of additional headers indicating email routing or forwarding.
 - Use of email authentication mechanisms (e.g., SPF, DKIM, DMARC).
7. **Behavioral Analysis:** Analysis of user behavior and interaction patterns with emails, such as:
 - User engagement metrics (e.g., open rate, click-through rate).
 - Frequency of marking emails as spam or moving them to spam folders.
 - Analysis of historical email interactions and user preferences.
8. **Machine Learning-Based Features:** Derived features generated through machine learning algorithms, such as:
 - Predicted probability scores from spam detection models.
 - Feature importance scores indicating the contribution of each feature to the classification decision.
 - Clustering or grouping of emails based on similarity in feature space.

1.4 Advantage

- Protection Against Malicious Activities:
- Enhanced Productivity:
- Improved User Experience
- Protection Against Offensive Content
- Reduced Risk of Security Breaches
- Preservation of Network Bandwidth
- Compliance with Regulations:
- Cost Savings

1.5 Scope

- It provides sensitivity to the client and adapts well to the
- Future spam techniques
- It considers a complete message instead of single words with
- Respect to its organization
- It increases security and control
- It reduces IT administration costs
- It also reduce Network Resource costs

1.3 Future work

The future work of email spam detection will likely focus on addressing emerging challenges and leveraging advanced technologies to improve detection accuracy, efficiency, and user experience. Here are some potential areas of future research and development

- **Deep Learning Techniques:**
- **Unsupervised Learning Approaches**
- **Multi-Modal Analysis**
- **Contextual Analysis**
- **Adversarial Defense Mechanisms:**
- **Privacy-Preserving Techniques**
- **Real-Time Feedback Loops:**
- **Cross-Platform Integration**
- **Explainable AI (XAI):**

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1 Services Used

Email spam detection typically involves the utilization of various services, both standalone and integrated within larger email security solutions. Here are some key services commonly used for email spam detection

Email Authentication Service

URL and Domain Reputation Services

Threat Intelligence Feeds

Machine Learning and AI Services

Anomaly Detection Services

Reporting and Feedback Mechanisms

Cloud-Based Spam Filtering Services

Managed Security Services

Anti-Spam Filtering Services

2.2 Tools and Software used

Tools and software used of email spam detection

Cisco Email Security

Microsoft Exchange Online Protection (EOP)

SpamTitan

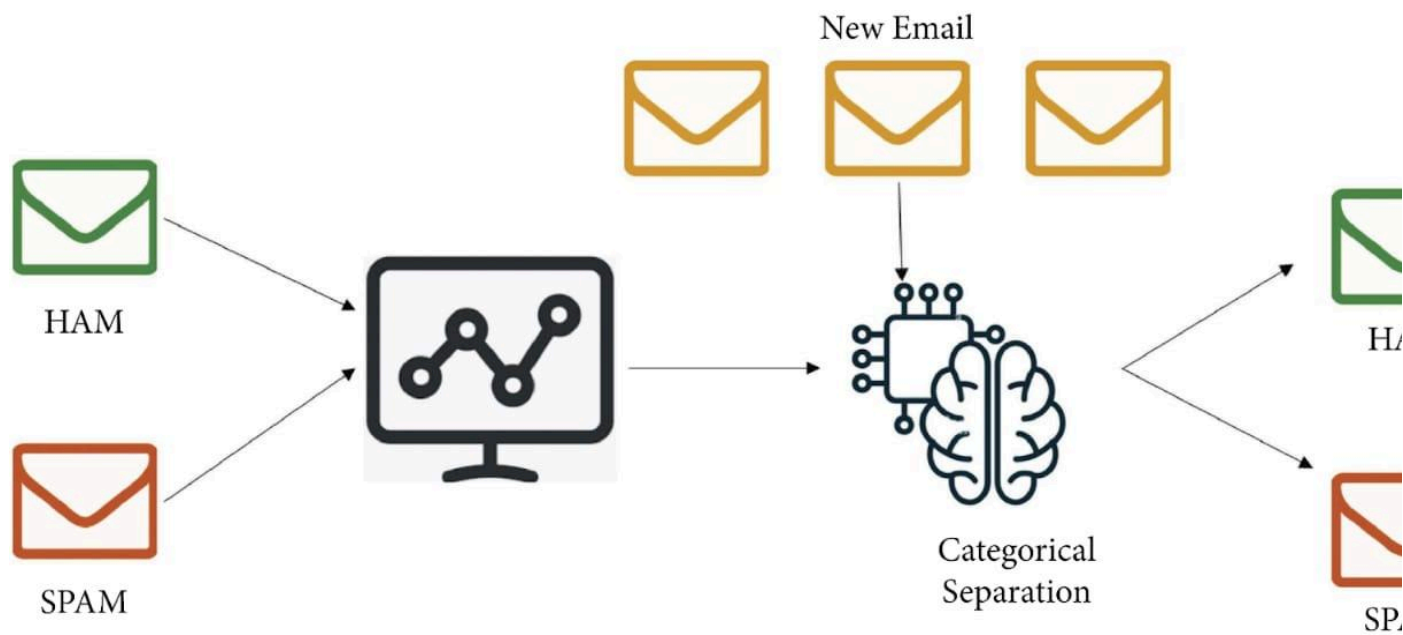
SpamAssassin

MailScanner

CHAPTER 3

PROJECT ARCHITECTURE

3.1 Architecture



CHAPTER 4

PROJECT OUTCOME

The project outcome of an email spam detection endeavor can vary depending on the specific goals, scope, and requirements of the project. However, here are some potential project outcomes that can be achieved:

1. **Development of a Functional Spam Detection System:** The primary outcome of the project may be the successful development and implementation of a functional email spam detection system. This system would be capable of automatically classifying incoming emails as either spam or legitimate based on various features and criteria.
2. **High Accuracy in Spam Detection:** The project outcome may include achieving high levels of accuracy in spam detection, as measured by metrics such as precision, recall, F1-score, and accuracy. A well-performing spam detection system should minimize false positives (legitimate emails classified as spam) and false negatives (spam emails classified as legitimate).
3. **Integration with Email Platforms:** The spam detection system may be integrated into email servers, clients, or filtering gateways to provide real-time protection against spam. Integration with existing email platforms ensures seamless operation and user accessibility.
4. **User-Friendly Interface:** The project may result in the development of a user-friendly interface that allows users to manage spam filtering preferences, view spam detection results, and provide feedback on detected emails. A intuitive interface enhances user experience and engagement with the spam detection system.
5. **Scalability and Efficiency:** The spam detection system should be scalable and efficient, capable of handling large volumes of incoming emails without significant performance degradation. Optimized algorithms and data processing techniques contribute to scalability and efficiency.
6. **Adaptability to New Threats:** The outcome may include mechanisms for continuous monitoring and adaptation to new spamming techniques and emerging threats. The spam detection system should be able to dynamically adjust its algorithms and criteria to effectively detect and mitigate evolving spam campaigns.
7. **Compliance with Regulations:** If applicable, the project outcome may involve ensuring compliance with relevant regulations and standards governing email communications and data privacy. This may include adherence to regulations such as the CAN-SPAM Act or GDPR.
8. **Documentation and Reporting:** Comprehensive documentation and reporting on the project outcomes, including details of the spam detection system architecture, algorithms used, performance metrics achieved, and user feedback. Clear documentation facilitates knowledge transfer and future maintenance of the system.

9. **Training and Support Materials:** Creation of training materials and user guides to assist users in understanding and effectively utilizing the spam detection system. Providing ongoing support and training ensures optimal use and adoption of the system.
10. **Evaluation and Validation:** The project outcome may include thorough evaluation and validation of the spam detection system's performance through testing, validation, and benchmarking against benchmark datasets or real-world email traffic. Validation ensures that the system meets the desired objectives and performance criteria.

```
import pandas as pd
```

Code cell <4Y2WaOc0EDuq>

```
# %% [code]
```

```
df = pd.read_csv('/content/archive.zip')
```

```
df
```

Execution output from Apr 20, 2024 12:59 PM

26KB

text/plain

	Address	Lot	AM or PM	\
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46	in	
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28	rn	
2	Unit 0065 Box 5052\nDPO AP 27450	94	vE	
3	7780 Julia Fords\nNew Stacy, WA 45798	36	vm	
4	23012 Munoz Drive Suite 337\nNew Cynthia, TX 5...	20	IE	
...		
9995	966 Castaneda Locks\nWest Juliafurt, CO 96415	92	XI	

9996 832 Curtis Dam Suite 785\nNorth Edwardburgh, T... 41 JY
 AM
 9997 Unit 4434 Box 6343\nDPO AE 28026-0283 74 Zh
 AM
 9998 0096 English Rest\nRoystad, IA 12457 74 cL
 PM
 9999 40674 Barrett Stravenue\nGrimesville, WI 79682 64 Hr
 AM

Browser Info \

0 Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...
 1 Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Pr...
 2 Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...
 3 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...
 4 Opera/9.58.(X11; Linux x86_64; it-IT) Presto/2...
 ...
 9995 Mozilla/5.0 (Windows NT 5.1) AppleWebKit/5352 ...
 9996 Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...
 9997 Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...
 9998 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_8;...
 9999 Mozilla/5.0 (X11; Linux i686; rv:1.9.5.20) Gec...

Date \	Company	Credit Card CC Exp
02/20	Martinez-Herman	6011929061123406
11/18	Fletcher, Richards and Whitaker	3337758169645356
08/19	Simpson, Williams and Pham	675957666125
02/24	Williams, Marshall and Buchanan	6011578504430710

10/25	4	Brown, Watson and Andrews	6011456623207998
...
03/22	9995	Randall-Sloan	342945015358701
07/25	9996	Hale, Collins and Wilson	210033169205009
05/21	9997	Anderson Ltd	6011539787356311
11/17	9998	Cook Inc	180003348082930
02/19	9999	Greene Inc	4139972901927273

	CC Security Code	CC Provider \
0	900	JCB 16 digit
1	561	Mastercard
2	699	JCB 16 digit
3	384	Discover
4	678	Diners Club / Carte Blanche
...
9995	838	JCB 15 digit
9996	207	JCB 16 digit
9997	1	VISA 16 digit
9998	987	American Express
9999	302	JCB 15 digit

	Email	
Job \		
0	pdunlap@yahoo.com	Scientist, product/process development

1	anthony41@reed.com	
Drilling engineer		
2	amymiller@morales-harrison.com	Customer
service manager		
3	brent16@olson-robinson.info	
Drilling engineer		
4	christopherwright@gmail.com	
Fine artist		
...	...	
...		
9995	iscott@wade-garner.com	
Printmaker		
9996	mary85@hotmail.com	
Energy engineer		
9997	tyler16@gmail.com	
Veterinary surgeon		
9998	elizabethmoore@reid.net	Local
government officer		
9999	rachelford@vaughn.com	
Embryologist, clinical		

	IP Address	Language	Purchase Price
0	149.146.147.205	el	98.14
1	15.160.41.51	fr	70.73
2	132.207.160.22	de	0.95
3	30.250.74.19	es	78.04
4	24.140.33.94	es	77.82
...
9995	29.73.197.114	it	82.21
9996	121.133.168.51	pt	25.63
9997	156.210.0.254	el	83.98
9998	55.78.26.143	es	38.84

```
9999 176.119.198.199 el 67.59

[10000 rows x 14 columns]

Code cell <87OKK39ME87S>
# %% [code]
df.head(10)
Execution output from Apr 20, 2024 12:59 PM
23KB
text/plain
    Address      Lot AM or PM \
0  16629 Pace Camp Apt. 448\nAlexisborough, NE 77... 46 in
PM
1  9374 Jasmine Spurs Suite 508\nSouth John, TN 8... 28 rn
PM
2              Unit 0065 Box 5052\nDPO AP 27450 94 vE
PM
3              7780 Julia Fords\nNew Stacy, WA 45798 36 vm
PM
4  23012 Munoz Drive Suite 337\nNew Cynthia, TX 5... 20 IE
AM
5  7502 Powell Mission Apt. 768\nTravisland, VA 3... 21 XT
PM
6      93971 Conway Causeway\nAndersonburgh, AZ 75107 96 Xt
AM
7  260 Rachel Plains Suite 366\nCastroberg, WV 24... 96 pG
PM
8              2129 Dylan Burg\nNew Michelle, ME 28650 45 JN
PM
9      3795 Dawson Extensions\nLake Tinafort, ID 88739 15 Ug
AM
```


Browser Info \

```

0 Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...
1 Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Pr...
2 Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...
3 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...
4 Opera/9.58.(X11; Linux x86_64; it-IT) Presto/2...
5 Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_8_5...
6 Mozilla/5.0 (compatible; MSIE 7.0; Windows NT ...
7 Mozilla/5.0 (X11; Linux i686) AppleWebKit/5350...
8 Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...
9 Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gec...

```

Company Credit Card CC Exp Date

```

0 Martinez-Herman 6011929061123406 02/20
1 Fletcher, Richards and Whitaker 3337758169645356 11/18
2 Simpson, Williams and Pham 675957666125 08/19
3 Williams, Marshall and Buchanan 6011578504430710 02/24
4 Brown, Watson and Andrews 6011456623207998 10/25
5 Silva-Anderson 30246185196287 07/25
6 Gibson and Sons 6011398782655569 07/24
7 Marshall-Collins 561252141909 06/25
8 Galloway and Sons 180041795790001 04/24
9 Rivera, Buchanan and Ramirez 4396283918371 01/17

```

CC Security Code CC Provider \

```

0 900 JCB 16 digit
1 561 Mastercard
2 699 JCB 16 digit

```

3	384	Discover
4	678	Diners Club / Carte Blanche
5	7169	Discover
6	714	VISA 16 digit
7	256	VISA 13 digit
8	899	JCB 16 digit
9	931	American Express
Email		
Job \		
0	pdunlap@yahoo.com	Scientist, product/process development
1	anthony41@reed.com	Drilling engineer
2	amymiller@morales-harrison.com	Customer service manager
3	brent16@olson-robinson.info	Drilling engineer
4	christopherwright@gmail.com	Fine artist
5	ynguyen@gmail.com	Fish farm manager
6	olivia04@yahoo.com	Dancer
7	phillip48@parks.info	Event organiser
8	kdavis@rasmussen.com	Financial manager
9	qcoleman@hunt-huerta.com	Forensic scientist
IP Address Language Purchase Price		
0	149.146.147.205	el 98.14

```
1      15.160.41.51      fr      70.73
2      132.207.160.22    de      0.95
3      30.250.74.19     es      78.04
4      24.140.33.94     es      77.82
5      55.96.152.147    ru      25.15
6      127.252.144.18    de      88.56
7      224.247.97.150    pt      44.25
8      146.234.201.229    ru      59.54
9      236.198.199.8     zh      95.63

Code cell <S1jlztauFWMR>

# %% [code]

df.tail(10)

Execution output from Apr 20, 2024 12:59 PM

22KB

text/plain

      Address      Lot AM or PM \
9990  75731 Molly Springs\nWest Danielle, VT 96934-5102  93 ty
PM
9991                PSC 8165, Box 8498\nAPO AP 60327-0346  50 dA
AM
9992  885 Allen Mountains Apt. 230\nWallhaven, LA 16995  40 vH
PM
9993  7555 Larson Locks Suite 229\nEllisburgh, MA 34...  72 jg
PM
9994                6276 Rojas Hollow\nLake Louis, WY 56410-7837  93 Ex
PM
9995                966 Castaneda Locks\nWest Juliafurt, CO 96415  92 XI
PM
9996  832 Curtis Dam Suite 785\nNorth Edwardburgh, T...  41 JY
AM
```

9997 Unit 4434 Box 6343\nDPO AE 28026-0283 74 Zh
 AM
 9998 0096 English Rest\nRoystad, IA 12457 74 cL
 PM
 9999 40674 Barrett Stravenue\nGrimesville, WI 79682 64 Hr
 AM

Browser Info \

9990 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4;...
 9991 Mozilla/5.0 (compatible; MSIE 8.0; Windows NT ...
 9992 Mozilla/5.0 (Macintosh; PPC Mac OS X 10_6_5) A...
 9993 Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_8...
 9994 Opera/9.68.(X11; Linux x86_64; sl-SI) Presto/2...
 9995 Mozilla/5.0 (Windows NT 5.1) AppleWebKit/5352 ...
 9996 Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...
 9997 Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...
 9998 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_8;...
 9999 Mozilla/5.0 (X11; Linux i686; rv:1.9.5.20) Gec...

	Company	Credit Card CC	Exp Date
9990	Pace, Vazquez and Richards	869968197049750	04/24
9991	Snyder Inc	4221582137197481	02/24
9992	Wells Ltd	4664825258997302	10/20
9993	Colon and Sons	30025560104631	10/25
9994	Ritter-Smith	3112186784121077	01/25
9995	Randall-Sloan	342945015358701	03/22
9996	Hale, Collins and Wilson	210033169205009	07/25
9997	Anderson Ltd	6011539787356311	05/21
9998	Cook Inc	180003348082930	11/17

9999	Greene Inc	4139972901927273	02/19
	CC Security Code	CC Provider	
Email \			
9990	877	JCB 15 digit	
andersonmichael@sherman.biz			
9991	969	Voyager	
king@wise-liu.com			
9992	431	Discover	
bberry@wright.net			
9993	629	Maestro	
chelseawilliams@lopez.biz			
9994	1823	Maestro	
iroberts@gmail.com			
9995	838	JCB 15 digit	
iscott@wade-garner.com			
9996	207	JCB 16 digit	
mary85@hotmail.com			
9997	1	VISA 16 digit	
tyler16@gmail.com			
9998	987	American Express	
elizabethmoore@reid.net			
9999	302	JCB 15 digit	
rachelford@vaughn.com			
	Job	IP Address	Language
Purchase Price			
9990	Early years teacher	54.170.3.185	ru
18.35			
9991	IT sales professional	254.25.31.156	el
25.93			
9992	Set designer	174.173.51.32	de
67.96			

65.61	9993	Designer, exhibition/display	177.46.82.128	el
31.85	9994	Education officer, museum	242.44.112.18	zh
82.21	9995	Printmaker	29.73.197.114	it
25.63	9996	Energy engineer	121.133.168.51	pt
83.98	9997	Veterinary surgeon	156.210.0.254	el
38.84	9998	Local government officer	55.78.26.143	es
67.59	9999	Embryologist, clinical	176.119.198.199	el
Code cell <3VzAu3NhFctO>				
# %% [code]				
df.dtypes				
Execution output from Apr 20, 2024 12:59 PM				
1KB				
text/plain				
	Address	object		
	Lot	object		
	AM or PM	object		
	Browser Info	object		
	Company	object		
	Credit Card	int64		
	CC Exp Date	object		
	CC Security Code	int64		
	CC Provider	object		
	Email	object		

```
Job                object

IP Address         object

Language           object

Purchase Price     float64

dtype: object

Code cell <4kRh1HwhFklA>

# %% [code]

df.isnull().sum()

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

Address           0

Lot               0

AM or PM          0

Browser Info      0

Company           0

Credit Card       0

CC Exp Date       0

CC Security Code  0

CC Provider       0

Email             0

Job               0

IP Address        0

Language          0

Purchase Price    0

dtype: int64

Code cell <Puw3FdOWFvwW>
```

```
# %% [code]
```

```
len(df.columns)
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

14

Code cell <FZ4aC5uRF_UQ>

```
# %% [code]
```

```
len(df)
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

10000

Code cell <Sx62YDL8GErF>

```
# %% [code]
```

```
df.info()
```

Execution output from Apr 20, 2024 1:00 PM

1KB

Stream

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 10000 entries, 0 to 9999

Data columns (total 14 columns):

#	Column	Non-Null Count	Dtype
0	Address	10000 non-null	object
1	Lot	10000 non-null	object
2	AM or PM	10000 non-null	object


```

3   Browser Info      10000 non-null object
4   Company           10000 non-null object
5   Credit Card       10000 non-null int64
6   CC Exp Date       10000 non-null object
7   CC Security Code  10000 non-null int64
8   CC Provider       10000 non-null object
9   Email             10000 non-null object
10  Job               10000 non-null object
11  IP Address        10000 non-null object
12  Language          10000 non-null object
13  Purchase Price    10000 non-null float64

dtypes: float64(1), int64(2), object(11)

memory usage: 1.1+ MB

```

Code cell <VMtrxIM1GJ1I>

```
# %% [code]
```

```
df.columns
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

```

Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company',
'Credit Card',
      'CC Exp Date', 'CC Security Code', 'CC Provider',
'Email', 'Job',
      'IP Address', 'Language', 'Purchase Price'],
      dtype='object')

```

Code cell <J8LbEhVpGhLD>

```
# %% [code]
```

```
df['Purchase Price'].max()
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

99.99

Code cell <ASl0WNTxHCsT>

%% [code]

df['Purchase Price'].min()

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

0.0

Code cell <0fj9MwJCHRp2>

%% [code]

df['Purchase Price'].mean()

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

50.347302

Code cell <q8d5tyc0Hc5C>

%% [code]

df.columns

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company',
'Credit Card',

```

        'CC Exp Date', 'CC Security Code', 'CC Provider',
        'Email', 'Job',

        'IP Address', 'Language', 'Purchase Price'],

        dtype='object')

```

Code cell <esOY5-8-HmDS>

```
# %% [code]
```

```
df['Language']=='fr'
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

```
0      False
```

```
1       True
```

```
2      False
```

```
3      False
```

```
4      False
```

```
...
```

```
9995    False
```

```
9996    False
```

```
9997    False
```

```
9998    False
```

```
9999    False
```

```
Name: Language, Length: 10000, dtype: bool
```

Code cell <Q-3fffM-HwNq>

```
# %% [code]
```

```
len(df[df['Language']=='fr'])
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

1097

Code cell <kybgEUgjIE3>

%% [code]

```
df[df['Language']=='fr'].count()
```

Execution output from Apr 20, 2024 1:00 PM

1KB

text/plain

Address	1097
---------	------

Lot	1097
-----	------

AM or PM	1097
----------	------

Browser Info	1097
--------------	------

Company	1097
---------	------

Credit Card	1097
-------------	------

CC Exp Date	1097
-------------	------

CC Security Code	1097
------------------	------

CC Provider	1097
-------------	------

Email	1097
-------	------

Job	1097
-----	------

IP Address	1097
------------	------

Language	1097
----------	------

Purchase Price	1097
----------------	------

dtype: int64

Code cell <hSEwKEPgIT-0>

%% [code]

```
df.columns
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company',  
'Credit Card',  
      'CC Exp Date', 'CC Security Code', 'CC Provider',  
'Email', 'Job',  
      'IP Address', 'Language', 'Purchase Price'],  
      dtype='object')
```

Code cell <ARLft1GuIa4e>

```
# %% [code]
```

```
len(df[df['Job'].str.contains('engineer',case=False)])
```

Execution output from Apr 20, 2024 1:00 PM

0KB

text/plain

984

Code cell <EDmqWDciJOEP>

```
# %% [code]
```

```
df.columns
```

Execution output from Apr 20, 2024 1:01 PM

0KB

text/plain

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company',  
'Credit Card',  
      'CC Exp Date', 'CC Security Code', 'CC Provider',  
'Email', 'Job',  
      'IP Address', 'Language', 'Purchase Price'],  
      dtype='object')
```

Code cell <m4Zalu90JU77>

```
# %% [code]
```

```
df[df['IP Address']=="132.207.160.22"]['Email']
```

Execution output from Apr 20, 2024 1:01 PM

0KB

text/plain

```
2      amymiller@morales-harrison.com
```

Name: Email, dtype: object

Code cell <NG88q-RJD2k0>

```
# %% [code]
```

```
len(df[(df['CC Provider']=="MasterCard") & (df['Purchase Price']>50)])
```

Execution output from Apr 20, 2024 1:01 PM

0KB

text/plain

```
0
```

Code cell <a2j5EPv6J2Ac>

```
# %% [code]
```

```
df[(df['CC Provider']=="Mastercard") \
    & (df['Purchase Price']>50)].count()
```

Execution output from Apr 20, 2024 1:01 PM

0KB

text/plain

```
Address      405
```

```
Lot          405
```

```
AM or PM     405
```

```
Browser Info 405
```

```
Company      405
```

```
Credit Card      405
CC Exp Date      405
CC Security Code  405
CC Provider      405
Email            405
Job              405
IP Address       405
Language         405
Purchase Price   405
dtype: int64
```

Code cell <hXzJ3YUWKZ2c>

```
# %% [code]
```

```
df[df['Credit Card']==4664825258997302]['Email']
```

Execution output from Apr 20, 2024 1:01 PM

0KB

text/plain

```
9992      bberry@wright.net
```

```
Name: Email, dtype: object
```

Code cell <c-yFukLXKlAk>

```
# %% [code]
```

```
df['AM or PM'].value_counts()
```

Execution output from Apr 20, 2024 1:01 PM

0KB

text/plain

```
AM or PM
```

```
PM      5068
```

```
AM      4932
```

```
Name: count, dtype: int64
```

```
Code cell <GhrdUDX6LjhM>
```

```
# %% [code]
```

```
df['CC Exp Date']
```

```
Execution output from Apr 20, 2024 1:01 PM
```

```
0KB
```

```
text/plain
```

```
0      02/20
```

```
1      11/18
```

```
2      08/19
```

```
3      02/24
```

```
4      10/25
```

```
...
```

```
9995    03/22
```

```
9996    07/25
```

```
9997    05/21
```

```
9998    11/17
```

```
9999    02/19
```

```
Name: CC Exp Date, Length: 10000, dtype: object
```

```
Code cell <OPySwPYALvTv>
```

```
# %% [code]
```

```
def fun():
```

```
    count=0
```

```
    for date in df['CC Exp Date']:
```

```
        if date.split('/')[1]=='20':
```

```
            count=count+1
```

```
    print(count)
```


Code cell <CXaujC7tMz-x>

```
# %% [code]
```

```
fun()
```

Execution output from Apr 20, 2024 1:01 PM

0KB

Stream

988

Code cell <K2w5C6nZM2NR>

```
# %% [code]
```

```
len(df[df['CC Exp Date'].apply(lambda x:x [3:]=='20')])
```

Execution output from Apr 20, 2024 1:01 PM

0KB

text/plain

988

Code cell <6rBTv07vNXjg>

```
# %% [code]
```

```
list1=[]
```

```
for email in df['Email']:
```

```
    list1.append(email.split('@')[1])
```

Code cell <Ytg7P7PWNvAX>

```
# %% [code]
```

```
df['temp']=list1
```

Code cell <L6EqDwnkN1dD>

```
# %% [code]
```

```
df.head(1)
```

Execution output from Apr 20, 2024 1:01 PM

11KB

text/plain

	Address	Lot	AM or PM	\
--	---------	-----	----------	---

0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46	in	
---	---------------------------------------------------	----	----	--

PM

Browser Info

Company \

0	Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...
---	---------------------------------------------------

Martinez-Herman

	Credit Card	CC	Exp Date	CC Security Code	CC Provider
--	-------------	----	----------	------------------	-------------

\

0	6011929061123406		02/20	900	JCB 16 digit
---	------------------	--	-------	-----	--------------

Email

Job

IP Address \

0	pdunlap@yahoo.com	Scientist, product/process development
---	-------------------	----------------------------------------

149.146.147.205

	Language	Purchase Price	temp
--	----------	----------------	------

0	el	98.14	yahoo.com
---	----	-------	-----------

Code cell <np5SwK-dN6e6>

```
# %% [code]
```

```
df['temp'].value_counts().head()
```

Execution output from Apr 20, 2024 1:02 PM

0KB

text/plain

```
temp

hotmail.com      1638

yahoo.com        1616

gmail.com        1605

smith.com         42

williams.com      37

Name: count, dtype: int64
```

Code cell <77sJk4q10Jm8>

```
# %% [code]
```

```
df['Email'].apply(lambda x:x.split('@')[1]).value_counts().head()
```

Execution output from Apr 20, 2024 1:02 PM

0KB

```
text/plain
```

```
Email

hotmail.com      1638

yahoo.com        1616

gmail.com        1605

smith.com         42

williams.com      37

Name: count, dtype: int64
```

CONCLUSION

Email has been the most important medium of communication nowadays, through internet connectivity any message can be delivered to all over the world. More than 270 billion emails are exchanged daily, about 57% of these are just spam emails. Spam emails, also known as non-self, are undesired commercial or malicious emails, which affects or hacks personal information like bank ,related to money or anything that causes destruction to single individual or a corporation or a group of people. Besides advertising, these may contain links to phishing or malware hosting websites set up to steal confidential information. Spam is a serious issue that is not just annoying to the end-users but also financially damaging and a security risk. Hence this system is designed in such a way that it detects unsolicited and unwanted emails and prevents them hence helping in reducing the spam message which would be of great benefit to individuals as well as to the company .In the future this system can be implemented by using different algorithms and also more features can be added to the existing system.

FUTURE SCOPE

The future scope of email spam detection is wide-ranging and holds considerable potential for innovation and improvement. Here are several areas that represent promising directions for future development

- Enhanced Accuracy with AI and Machine Learning
- Behavioral Analysis and Contextual Understanding
- Multi-Modal Analysis
- Real-Time Threat Intelligence and Collaboration
- Privacy-Preserving Techniques
- Cross-Platform Integration
- Adaptive and Self-Learning Systems
- Explainable AI (XAI)

REFERENCES

- [1] S. H. a. M. A. T. Toma, "An Analysis of Supervised Machine Learning Algorithms for Spam Email Detection," in International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021.

- [2] S. Nandhini and J. Marseline K.S., "Performance Evaluation of Machine Learning Algorithms for Email Spam Detection," in International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020.

- [3] A. L. a. S. S. S. Gadde, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," in 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, 2021.

- [4] V. B. a. B. K. P. Sethi, "SMS spam detection and comparison of various machine learning algorithms," in International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), 2017.

- [5] G. D. a. A. R. P. Navaney, "SMS Spam Filtering Using Supervised Machine Learning Algorithms," in 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018

CODE

Please Provide Code through Git Hub Repo Link

<https://github.com/Srirameee21/codessriram>

VIDEO IN GITHUB LINK

**[https://github.com/Srirameee21/codessriram/blob/main/README.
md](https://github.com/Srirameee21/codessriram/blob/main/README.md)**

ALGORITHM LINK IN GITHUB

**[https://github.com/Srirameee21/codessriram/blob/main/email%20
b%20sriram.pdf](https://github.com/Srirameee21/codessriram/blob/main/email%20b%20sriram.pdf)**

PPT LINK IN GITHUB

**[https://github.com/Srirameee21/codessriram/blob/main/PPT_srira
m%20b.pptx](https://github.com/Srirameee21/codessriram/blob/main/PPT_sriram%20b.pptx)**